New Tools and Results in Graph Structure Theory

A Thesis
Presented to
The Academic Faculty

by

Rajneesh Hegde

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Mathematics
Georgia Institute of Technology
May 2006

New Tools and Results in Graph Structure Theory

Approved by:

Prof. Robin Thomas, Advisor
School of Mathematics
Georgia Institute of Technology

Prof. Richard Duke
School of Mathematics
Georgia Institute of Technology

Prof. Xingxing Yu
School of Mathematics
Georgia Institute of Technology

Prof. Prasad Tetali
School of Mathematics
Georgia Institute of Technology

Prof. Bill Cook
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: February 23, 2006

The desire to be right and the desire to have been right are two desires, and the sooner we separate them the better off we are. The desire to be right is the thirst for truth. On all accounts, both practical and theoretical, there is nothing but good to be said for it. The desire to have been right, on the other hand, is the pride that goeth before a fall. It stands in the way of our seeing we were wrong, and thus blocks the progress of our knowledge.

W. V. Quine and J. S. Ullian, *The Web of Belief*

To my parents...

They have always put their children's interests ahead of their own, and have given me more love and support than I could have asked for, or deserved.

ACKNOWLEDGEMENTS

I wish to thank my advisor, Prof. Robin Thomas, for his guidance and direction throughout my stay at Georgia Tech. I also wish to thank Professors Evans Harrell and Luca Dieci for their help and support as graduate coordinators for the School of Mathematics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

We first prove a "non-embeddable extensions" theorem for polyhedral graph embeddings. Let G be a "weakly 4-connected" planar graph. We describe a set of constructions that produce a finite list of non-planar graphs, each having a minor isomorphic to G, such that every non-planar weakly 4-connected graph H that has a minor isomorphic to G has a minor isomorphic to one of the graphs in the list. The theorem is more general and applies in particular to polyhedral embeddings in any surface.

We discuss an approach to proving Jorgensen's conjecture, which states that if G is a 6-connected graph with no K_6 minor, then it is apex, that is, it has a vertex v such that deleting v yields a planar graph. We relax the condition of 6-connectivity, and prove Jorgensen's conjecture for a certain sub-class of these graphs.

We prove that every graph embedded in the Klein bottle with representativity at least 4 has a K_6 minor. Also, we prove that every "locally 5-connected" triangulation of the torus, with one exception, has a K_6 minor. (Local 5-connectivity is a natural notion of local connectivity for a surface embedding.) The above theorem uses a locally 5-connected version of the well-known splitter theorem for triangulations of any surface.

We conclude with a theoretically optimal algorithm for the following graph connectivity problem. A shredder in an undirected graph is a set of vertices whose removal results in at least three components. A 3-shredder is a shredder of size three. We present an algorithm that, given a 3-connected graph, finds its 3-shredders in time proportional to the number of vertices and edges, when implemented on a RAM (random access machine).

CHAPTER I

INTRODUCTION

This chapter is intended to be both a brochure and a map. In Section 1.1 (the brochure part), I have attempted to give a brief yet (hopefully) useful overview of the theme of this thesis. It is aimed primarily at the reader who is not already familiar with the particular neighborhood of discrete mathematics that my research inhabits, and it highlights some of the most important results in that theory. Section 1.2 (the map part) briefly describes the results in this thesis, along with the larger context that surrounded and motivated it. It is intended to serve as a navigational aid for the rest of the thesis.

1.1 The What and the Why

Most of the research described in this thesis relates, directly or indirectly, to *Graph Structure Theory*. Before diving into it, a few words about what graphs are, and what they are good for, might be in order.

A *graph* can be defined abstractly as a pair of sets: a (finite, non-empty) set of *vertices* (or points), and a set of unordered pairs of vertices, called *edges* (or lines). An edge is said to join the two vertices specified in the pair, and the two vertices are said to be *adjacent*. (As far as a mental picture is concerned, it is almost always more fruitful to think of graphs as consisting of points, and lines joining those points.) Unless otherwise specified, the graphs that we consider are *simple graphs*, that is, they do not have two edges sharing both end-points, or an edge with coincident end-points.

It should not take much imagination to see why graphs are useful: they can model any sort of binary relationship among a set of objects. The objects themselves can be represented by vertices in a graph, and related pairs of objects give rise to edges. Thus the Internet is really a graph in disguise, and so is a family tree, or a map of the world. (For the latter example, think of the countries as the vertices, with edges joining pairs of countries that share a border.)

1.1.1 Graph Structure Theory

Graph Structure Theory is the area of Graph Theory that concerns itself primarily with questions of the following two flavors:

1. When does some graph contain a certain other graph as a "substructure"?

2. How does having certain substructures affect other (and seemingly unrelated) properties of the graph?

Questions of the first kind are perhaps best motivated by an appeal to our inclination towards modularity. We like our problems, and solutions, to come in bite-sized pieces. The best way to understand a complex thing is usually to break it down into smaller components, and try to understand the components; graphs are no exception. The theorems in Chapters 3 and 5, for instance, illustrate problems of this nature.

As for questions of the second kind, there are two reasons to study them, one theoretical and one practical. From a historical standpoint, surprising connections between seemingly unrelated problems have always been of interest to mathematicians. On a more practical note, if some graph property has a structural characterization, it usually leads to an efficient algorithm to recognize it. For instance, the algorithm presented in Chapter 6 is motivated by such a structural characterization.

A good example of the algorithmic contribution of Graph Structure Theory is the use of "tree decompositions" in efficiently solving optimization problems that are intractable in general. Roughly speaking, a tree decomposition breaks a graph into pieces joined along a tree-like structure. If the pieces in this structure are of bounded size ("tree-width"), then certain traditionally hard problems can be solved for these graphs using a hierarchical (bottom-up) approach [40, 6]. Even when the tree-width is not bounded, these decompositions are useful in obtaining efficient approximation algorithms [18, 9].

Now that we agree that the above kinds of problems are worthwhile endeavors, the question is, what do we mean by a "substructure"? The substructure relation that has proven most fruitful in this context, by far, is that of a *graph minor*. We say that a graph G is a *minor* of a graph H if the vertices of G are in one-to-one correspondence with disjoint connected subgraphs of H, in such a way that adjacencies (in G) are preserved: that is, for every edge in G, there is an edge between the corresponding subgraphs of H. (Another way of thinking of a graph minor is the following: G is a minor of H if G can be obtained from H by "contracting" or deleting edges, or deleting vertices. Contracting an edge[1] means identifying its end-points u and v and joining the new vertex to any vertex that was adjacent to u or v, or both.)

The theory of graph minors is very rich, with theorems that have deep implications in graph theory and elsewhere. The most important result in this theory, and perhaps the deepest theorem in graph theory, is the following theorem of Robertson and Seymour:

Theorem 1.1.1 (Graph Minor Theorem, formerly Wagner's conjecture**)** *Given any infinite sequence $G_1, G_2, \ldots$ of (finite) graphs, there exist integers $i < j$ such that G_i is a minor of G_j.*

The original proof of the theorem appeared in a series of over twenty papers under the common title *Graph Minors*, most of which were published in the *Journal of Combinatorial Theory, Series B*. Simpler proofs have since been found for some parts of that proof. For a good overview, refer to [10].

Like most fundamental theorems in mathematics, the proof of the Graph Minor Theorem has spawned numerous techniques that are interesting in their own right. However, one particular implication of the theorem stands out for its far-reaching consequences in graph theory and the complexity of graph algorithms. Let $\mathcal{F}$ be a family of graphs that is minor-closed (that is, if $H \in \mathcal{F}$ and G is a minor of H, then $G \in \mathcal{F}$). It is an easy exercise to show that $\mathcal{F}$ can then be characterized in terms of a set of *forbidden* (or *excluded*) minors. In other words, a graph G is a member of $\mathcal{F}$ if and only if none of the forbidden graphs is a minor of G. From the Graph Minor Theorem, we can conclude that in fact the set of forbidden minors can be chosen to be finite.

[1] There are minor variations of this definition, depending on the context in which graph minors are used.

Now, for a *fixed* graph X (not part of the input), there is an $O(n^3)$-time algorithm to test if an input graph G has a minor isomorphic to X (where n is the number of vertices). Thus it immediately follows that there exists an $O(n^3)$-time membership test for *any* minor-closed family. (However, the list of forbidden minors, and the algorithm, are only given implicitly.)

Corollary 1.1.2 *Given any minor-closed family $\mathcal{F}$ of graphs, there is a finite set of graphs such that a graph G is a member of $\mathcal{F}$ if and only if no graph in this set is a minor of G.*

Among the more dramatic algorithmic consequences of the kind stated above is for the problem of deciding whether a graph is *knotless* (that is, embeddable in 3-space such that none of its cycles forms a non-trivial knot). Before the Graph Minor Theorem came along, it was not even known whether the problem was decidable, that is whether *any* algorithm, however slow, existed. To this day, no explicit algorithm for this problem is known. However, the family of knotless graphs is minor-closed, so the above corollary immediately implies the existence of a polynomial, indeed $O(n^3)$-time, algorithm.

Minor-closed graph families are quite common in graph theory. Among the most natural (and historically significant) of these is the family of graphs embeddable in a given surface. Thus another powerful corollary of the Graph Minor Theorem is the following:

Corollary 1.1.3 *For any surface Σ, there is a finite list of forbidden minors, such that a graph is embeddable in Σ if and only if it has none of the forbidden minors.*

The prototypical theorem of the above kind, indeed one that is a precursor of Graph Structure Theory itself, is *Kuratowski's Theorem* [30], which states that a graph can be embedded in the plane (with no edge crossings) if and only if it does not have a minor isomorphic to either of two graphs, K_5 and $K_{3,3}$. (The former is the complete graph on five vertices, while the latter is a graph consisting of six vertices, three in each of two classes, with all possible edges between the classes.) Thus the above corollary is a kind of "generalized Kuratowski theorem". It had been conjectured in the 1930s by Erdős and König. It was verified for non-orientable surfaces in [4], and for orientable surfaces in [5].

Robertson and Seymour [44] gave a proof for general surfaces before their proof of the Graph Minor Theorem itself. Recently, Thomassen gave a simpler proof for the result in [56].

A major ingredient in the proof of the Graph Minor Theorem, and indeed a deep result in its own right, is the *Structure Theorem*, which gives for any integer n a structural description of the graphs that do not have a minor isomorphic to the complete graph K_n. The statement of the theorem is somewhat technical, and we will not state it precisely here. Very roughly speaking, however, it says that every graph not having a K_n-minor is a subgraph of a graph obtained by pasting together "nearly embeddable" graphs in a surface in which K_n itself is not embeddable, where the pasting is along complete subgraphs of a size bounded above by a number depending only on n (and not the size of the graph itself).

Yet another deep result in the Robertson-Seymour theory is a polynomial-time algorithm for the *Disjoint Rooted Paths* problem stated below:

Instance: A graph G and disjoint subsets $S = \{s_1, \ldots, s_k\}$ and $T = \{t_1, \ldots, t_k\}$ of "terminals" (vertices), where k is a *fixed* integer (not part of the input).

Question: Does G have disjoint paths joining s_i and t_i, for $i = 1, \ldots, k$?

Before its resolution, the above was a significant open problem in the theory of graph algorithms, and its complexity status was unknown. As with the Graph Minor Theorem, however, the significance of the algorithm lies as much in its repercussions as in the result itself.

The case $k = 2$ is particularly nice, with the following beautiful structural characterization for the existence of the paths. A graph has two disjoint s_i-t_i paths (modulo some natural connectivity requirements) if and only if it cannot be drawn in the plane with the vertices s_1, s_2, t_1, t_2 on the infinite face, in that order. (It is trivial to check that the above condition is necessary.) The theorem was proved independently in [48], [49], and [54]. [49] gives an $O(nm)$-time algorithm that, given a graph with four terminals as above, either finds two paths as required, or produces a planar drawing demonstrating their infeasibility. (Here n and m are the numbers of vertices and edges respectively.) It should be noted that for the general k-paths problem above, no structural characterization is known for $k \geq 3$.

Graph coloring offers another tantalizing connection with Graph Structure Theory, with some deep questions yet to be answered. Consider the "chromatic number" of a graph, that is, the least number of colors that one can color the vertices with, such that no two adjacent vertices get the same color. When does a graph need, say, at least 10 colors? One natural possibility is when the graph has 10 mutually adjacent vertices. One can now reasonably ask whether containing such a "complete graph" (or clique) is the only reason for a high chromatic number. For the right notion of containment (that is, graph minors), this is in fact a beautiful and long-standing conjecture:

Conjecture 1.1.4 (Hadwiger's Conjecture) *For every integer t, if a graph has chromatic number $\geq t$, then it has a minor isomorphic to the complete graph K_t on t vertices.*

Hadwiger's conjecture is trivial for $t \leq 3$ and easy for $t = 4$, but things go steeply uphill from there. For $t = 5$, it implies the four-color theorem, a deep theorem which had to wait over a hundred years for a proof. In fact, it is equivalent to the four-color theorem, as shown by Wagner's structural characterization of graphs with no K_5 minor [59]. A deep theorem of Robertson, Seymour and Thomas [45] establishes the case $t = 6$; more precisely, it proves that Hadwiger's conjecture for $t = 6$ is also equivalent to the four-color theorem. The conjecture is open for all $t \geq 7$. (It is easy to show that the conjecture for $t + 1$ implies it for t.)

Hadwiger's conjecture has been shown to be true for "almost all" graphs. (More precisely: in $\mathcal{G}(n, p)$, a natural and well-studied probabilistic model for a random graph, the probability that a graph has a clique minor of size equal to its chromatic number has been shown to approach 1 as the number n of vertices tends to infinity.) Also, a deterministic, but approximate version of the conjecture is known: by a theorem independently proved in [29] and [53], a graph with chromatic number at least $ct\sqrt{\log t}$ has a K_t minor. It should be noted that this theorem obtains the minor using merely the corresponding lower bound on the average degree. The main question that Hadwiger's conjecture raises is whether the much stronger property of having a high chromatic number has a deeper connection to clique minors than the weaker property of having lots of edges.

1.2 Research Summary

1.2.1 Non-embeddable Extensions of Embedded Minors

A graph is a *subdivision* of another if the first can be obtained from the second by replacing each edge by a non-zero length path with the same ends, where the paths are disjoint, except possibly for shared ends. A graph G is said to be a *topological minor* of a graph H if a subdivision of G is a subgraph of H. (A topological minor is a special case of a minor.)

Suppose a non-planar graph H has a minor isomorphic to a planar graph G. For various problems in Graph Structure Theory it is useful to know the minor-minimal graphs that (i) are a minor of H, (ii) have a minor isomorphic to G, and (iii) are non-planar. In other words, one wants to know what more does H contain on account of its non-planarity.

In the applications of such a result, G is explicitly known, whereas H is not, and the enlargement operations would furnish an explicit list of graphs such that (i) H has a minor isomorphic to one of the graphs on the list, and (ii) each graph on the list is a witness both to the fact that G is isomorphic to a minor of H, and that H is, in addition, non-planar. (The minor-minimality of the graphs in the list is required to avoid redundancy — if G_1, G_2 are in the list with G_1 being a minor of G_2, then we could throw away G_2 from the list without affecting the validity of the previous statement.)

Under some mild connectivity assumptions, [41] finds these "minimal non-planar enlargements" of G when the containment relation is that of a topological minor. This theorem has been used in [11, 51], and will probably be useful elsewhere as well. However, in more complicated applications it is more efficient to work with minors, rather than topological minors.

Chapter 2 describes a theorem for finding non-planar extensions of planar minors. In fact, it turns out that such a tool can be formulated in terms of *disk systems*. (Chapter 2 gives a precise definition of a disk system. Roughly speaking, it is an abstract generalization of the set of facial cycles of a graph embedding.) In particular, the results of Chapter 2 apply to "polyhedral embeddings" in any surface, not just the plane.

[41] describes a related theorem for "non-apex extensions". A graph is said to be *apex* it has some vertex whose removal yields a planar graph. Thus, if an apex graph G is a minor

of a non-apex graph H, one would want to know the list of minimal non-apex extensions of G, such that H has a minor isomorphic to one of the graphs in the list.

Under connectivity assumptions similar to the ones in the above theorem, [41] gives a list of these non-apex extensions. A potential application of this result is described in Section 1.2.2.

1.2.2 Six-regular Graphs in $\mathcal{F}_6$

The proof in [45] establishing Hadwiger's conjecture for $t = 6$ considers a minor-minimal counterexample to the conjecture. In other words, it considers a graph that is minor-minimal subject to (i) having chromatic number ≥ 6, and (ii) having no K_6 minor. It then proves that such a graph must be apex. By the four-color theorem, it follows that the graph is 5-colorable, and thus not a counterexample after all. Mader [33] proved that a minor-minimal counterexample as above must be 6-connected. (A graph is k-connected if the removal of fewer than k vertices does not disconnect it.)

Jorgensen asked whether it is the 6-connectivity that is the crucial property here: more precisely, he made the following conjecture.

Conjecture 1.2.1 (Jorgensen's Conjecture, [24]) *Every 6-connected graph with no K_6 minor is apex.*

From the discussion in the previous paragraph, it follows that Jorgensen's conjecture immediately implies Hadwiger's conjecture for $t = 6$. Understanding the structure of graphs with no K_t minor is an important problem since excluded minors play an important role with regard to several fundamental graph properties. Moreover, Jorgensen's conjecture can be seen as a Kuratowski-like theorem for apex graphs — modulo 6-connectivity, graphs are apex if and only if they have no K_6 minor.(The forward implication is trivial: K_6 itself is non-apex since K_5 is non-planar, and thus a graph with a K_6 minor cannot be apex.)

The difficulty with trying to prove Jorgensen's conjecture, however, is that the hypothesis of 6-connectivity (or k-connectivity, in general) does not behave well under taking minors. One possible approach to getting around this problem is thus to consider a different notion of connectivity. Mader considered one such notion in [32], where he defined a family

$\mathcal{F}_5$ of graphs. A graph is in $\mathcal{F}_5$ if it has at least six vertices and has minimum degree five, *except* perhaps a set of mutually adjacent vertices of degree less than five. Thus being in $\mathcal{F}_5$ is a weaker property than 5-connectivity. The intuition is that this weaker condition is more robust under taking minors, and that one can thus say what the minimal graphs in the family look like. Indeed, [32] explicitly lists all the minimal graphs in $\mathcal{F}_5$ (there are four such graphs, two of which are K_6 and the graph of the icosahedron).

It is thus reasonable to ask whether an analogous weakening of 6-connectivity might be useful for attacking Jorgensen's conjecture. We thus define $\mathcal{F}_6$ as the family of graphs with at least seven vertices and with minimum degree six, except perhaps a set of mutually adjacent vertices of degree less than six. A counterexample G to the conjecture must have a minor isomorphic to a minimal member of $\mathcal{F}_6$. If that minimal graph itself has a K_6 minor, then we are done. If it is apex, however, then one could use the non-apex extensions theorem mentioned in Section 1.2.1 and verify whether applying them always leads to a K_6 minor.

The list of minor-minimal graphs in $\mathcal{F}_6$ is not known, and it seems that finding that list explicitly might involve a formidable effort. However, we verified that Jorgensen's conjecture does hold for a certain subfamily of $\mathcal{F}_6$:

Theorem 1.2.2 *Every minor-minimal graph in $\mathcal{F}_6$ with no low-degree vertices (that is, with minimum degree 6) has a K_6 minor.*

In particular, for this sub-class of $\mathcal{F}_6$, the implication in Jorgensen's conjecture holds vacuously (in the sense that the antecedent is never true). A proof of the above theorem is given in Chapter 3.

1.2.3 A Splitter Theorem for Triangulations

If G and H are connected graphs such that G is a minor of H, then H can be obtained from G by repeatedly "splitting" vertices and adding edges. (Vertex splits, as explained in Chapter 4, are inverses of the usual edge contractions, for edges not contained in triangles.) Seymour's well-known splitter theorem says that if G and H are both 3-connected, and G

is not a wheel, then the above sequence of operations can be chosen such that every intermediate graph is also 3-connected. This theorem has been very useful in various contexts [52].

In Chapter 4, we prove a splitter theorem for "locally 5-connected" triangulations of a general surface. In other words, (i) the edge contractions and vertex splits are defined in the context of surface triangulations, (ii) G and H are both "locally 5-connected" (a natural definition of local connectivity for a surface embedding). The theorem, then, states that H can be obtained from G by applying a sequence of vertex splits such that every intermediate graph is also locally 5-connected. (Since G and H are both triangulations of the same surface, it is easy to see that edge additions are not needed.) One application of this splitter theorem is in the proof of Theorem 5.1.2.

1.2.4 K_6 Minors in the Torus and the Klein Bottle

In Chapter 5, we prove two theorems about K_6 minors in graphs embedded in the torus or the Klein bottle. The first of these states that "locally 5-connected" triangulations in the torus, with one particular exception, have a minor isomorphic to the complete graph K_6. The second theorem states that every graph embedded in the Klein bottle with representativity at least four has a minor isomorphic to K_6.

While these theorems might be of independent interest, our primary motivation for proving them was their usefulness for Theorem 1.2.2. Towards the end of the proof of that theorem, we encounter a minimal graph in $\mathcal{F}_6$ that is a 6-regular triangulation of the torus or the Klein bottle. The theorems mentioned in the previous paragraph imply, in particular, that such a graph must indeed have a K_6 minor.

1.2.5 Finding 3-shredders Efficiently

A *shredder* in a graph is a set of vertices whose removal results in at least three connected components. (Thus a shredder is a special case of a vertex-cut.) A 3-*shredder* is a shredder of size three. In Chapter 6, we present an algorithm that, given a 3-connected graph, finds its 3-shredders in time proportional to the number of vertices and edges, when implemented on a RAM (random access machine).

The motivation to study this problem came from the even directed cycle problem. Given a digraph D, the question is to decide whether D has a directed cycle of even length. This is equivalent [58] to several other problems of interest, for instance: Given a 0-1 square matrix A, can some of the 1's be changed to -1's in such a way that the permanent of A equals the determinant of the modified matrix (Pólya's permanent problem, [38])? When does a bipartite graph have a "Pfaffian orientation" [26, 27])? When is a square matrix *sign-nonsingular* [7, 28]?

For the version above that is phrased in terms of Pfaffian orientations, [34, 35, 43] present an $O(n^3)$ algorithm, based on a structural characterization of bipartite graphs that possess a Pfaffian orientation. The exact definition of a Pfaffian orientation is not important here; what is relevant is that the structural characterization is in terms of a "trisum" operation that pastes *three* smaller graphs along a cycle of 4 vertices. With careful implementation, the running time of that algorithm can be reduced to $O(n^2)$, but attempts at further improvements run into serious difficulty. In order to take advantage of the structure theorem of [43], one needs to be able, at the very least, to efficiently decide whether a 4-connected bipartite graph has a 4-shredder. It is not clear to us whether the bipartite-ness would help. Given that the corresponding problem for 3-shredders in 3-connected graphs was not known, we started with that as the first step. It should be noted that the 4-connected graphs in the above application have $O(n)$ edges, so a linear ($O(n + m)$) running time would indeed be an improvement over $O(n^2)$, and is theoretically optimal.

1.2.6 A Note on Terminology

Most of the notation used in this thesis is standard to Graph Theory.

Specific definitions are usually made just before their use, instead of being collected in a common glossary. General or common terminology, on the other hand, is explained here or at the beginning of the respective chapters.

Unless otherwise stated, graphs are finite, undirected and simple (that is, with no parallel edges or loops).

For a graph G and an edge e in G, G/e denotes the graph obtained by *contracting* the edge e, that is, identifying its end-points and removing any resulting parallel edges. $G\backslash e$ denotes the graph obtained by deleting the edge e. (Similarly, if X is a subset of the vertex or edge sets, or is a subgraph of G, then $G\backslash X$ denotes the graph obtained by deleting the vertices and/or edges in X.)

A graph is a *minor* of another if the first can be obtained from a subgraph of the second by contracting edges. A graph is a *subdivision* of another if the first can be obtained from the second by replacing each edge by a non-zero length path with the same ends, where the paths are disjoint, except possibly for shared ends. The replacement paths are called *segments*, and their ends are called *branch-vertices*. A graph is a *topological minor* of another if a subdivision of the first is a subgraph of the second.

Paths and *cycles* have no "repeated" vertices. For $Z \subseteq V(G)$, $G[Z]$ denotes the subgraph *induced* by Z, that is, the subgraph consisting of Z and all edges with both ends in Z. A subgraph of G is said to be *induced* if it is induced by its vertex set.

A *separation* of a graph G is a pair (A, B) of subsets of $V(G)$ such that $A \cup B = V(G)$, and there is no edge between $A - B$ and $B - A$. The separation is called *non-trivial* or *proper* if both A and B are proper subsets of $V(G)$. The *order* of (A, B) is $|A \cap B|$. A separation of order k (respectively, of order $\leq k$) is called a k-separation (respectively, a $\leq k$-separation).

CHAPTER II

NON-EMBEDDABLE EXTENSIONS OF EMBEDDED MINORS

2.1 Introduction

Let a non-planar graph H have a subgraph isomorphic to a subdivision of a planar graph G. For various problems in Graph Structure Theory it is useful to know the minimal subgraphs of H that are isomorphic to a subdivision of G and are non-planar. In other words, one wants to know what more does H contain on account of its non-planarity. In [41] it is shown that under some mild connectivity assumptions these "minimal non-planar enlargements" of G are quite nice. In the applications of the result, G is explicitly known, whereas H is not, and the enlargement operations would furnish an explicit list of graphs such that (i) H has a subgraph isomorphic to a subdivision of one of the graphs on the list, and (ii) each graph on the list is a witness both to the fact that G is a topological minor of H, and that H is, in addition, non-planar. (The minimality of the graphs in the list is required to avoid redundancy.) Before we state that result, we need a few definitions.

A graph G is *weakly* 4*-connected* if G is 3-connected, has at least five vertices, and for every separation (A, B) of G of order at most three, one of the graphs $G[A]$, $G[B]$ has at most four edges.

A cycle C in a graph G is called *peripheral* if it is induced and $G\backslash V(C)$ is connected. It is well-known [57, 60] that the peripheral cycles in a 3-connected planar graph are precisely the cycles that bound faces in some (or, equivalently, every) planar embedding of G.

Let S be a subgraph of a graph H. An *S-path* in H is a path with both ends in S, and otherwise disjoint from S. Let C be a cycle in S, and let P_1 and P_2 be two disjoint S-paths in H with ends u_1, v_1 and u_2, v_2, respectively, such that u_1, u_2, v_1, v_2 belong to $V(C)$ and occur on C in the order listed. In those circumstances we say that the pair P_1, P_2 is an

S-cross in H. We also say that it is an *S-cross on C*. We say that u_1, v_1, u_2, v_2 are the *feet* of the cross. We say that the cross P_1, P_2 is *free* if

(F1) for $i = 1, 2$ no segment of S includes both ends of P_i, and

(F2) no two segments of S that share a vertex include all the feet of the cross.

The following was proved in [41].

Theorem 2.1.1 *Let G be a weakly 4-connected planar graph, and let H be a weakly 4-connected non-planar graph such that a subdivision of G is isomorphic to a subgraph of H. Then there exists a subgraph S of H isomorphic to a subdivision of G such that one of the following conditions holds:*

1. *there exists an S-path in H such that its ends belong to no common peripheral cycle in S, or*

2. *there exists a free S-cross in H on some peripheral cycle of S.*

This theorem has been used in [11, 51], and will probably be useful elsewhere as well. However, in more complicated applications it is more efficient to work with minors, rather than topological minors. We sketch one such application in Section 2.8. For any fixed graph G, there exists a finite and explicitly constructible set $\{G_1, G_2, \ldots, G_t\}$ of graphs such that a graph H has a minor isomorphic to G if and only if it has a topological minor isomorphic to one of the graphs G_i. Thus one can apply Theorem 2.1.1 t times to deduce the desired conclusion about G, but it would be nicer to have a more direct route to the result that involves less potential duplication. Furthermore, if the outcome is allowed to be a minor of H rather than a topological minor, then the outcomes (i) and (ii) above can be strengthened to require that the ends of the paths involved are branch-vertices of S, as we shall see.

It turns out that Theorem 2.1.1 is not exclusively about face boundaries of planar graphs, but that an appropriate generalization holds under more general circumstances. Thus rather than working with peripheral cycles in planar graphs we will introduce an appropriate set of axioms for a set of cycles of a general graph. We do so now in order to avoid having to restate our definitions later when we present the more general form of our results.

A *segment* in a graph G is a maximal path such that its internal vertices all have degree in G exactly two. If a graph G has no vertices of degree two, then the segments of a subdivision of G defined earlier coincide with the notion just defined. Since we will not consider subdivisions of graphs with vertices of degree two there is no danger of confusion. A *cycle double cover* in a graph G is a set $\mathcal{D}$ of distinct cycles of G, called *disks*, such that

(D1) each edge of G belongs to precisely two members of $\mathcal{D}$.

A cycle double cover $\mathcal{D}$ is called a *disk system* in G if

(D2) for every vertex v of G, the edges incident with v can be arranged in a cyclic order such that for every pair of consecutive edges in this order, there is precisely one disk in $\mathcal{D}$ containing that pair of edges, and

(D3) the intersection of any two distinct disks in $\mathcal{D}$ either has at most one vertex or is a segment.

A cycle double cover satisfying (D3) is called a *weak disk system.* It is easy to see that if a connected graph has a disk system, then it is a subdivision of a 3-connected graph. Also, note that in a 3-connected graph, Axiom (D3) is equivalent to the requirement that every two distinct disks intersect in a complete subgraph on at most two vertices. The peripheral cycles of a 3-connected planar graph form a disk system. More generally, if G is a subdivision of a 3-connected graph embedded in a surface Σ in such a way that every homotopically nontrivial closed curve intersects the graph at least three times (a "polyhedral embedding"), then the face boundaries of this embedding form a disk system in G. Conversely, it can be shown that a disk system in a graph is the set of face boundaries of a polyhedral embedding of the graph in some surface. Weak disk systems correspond to face boundaries of embeddings into pseudo-surfaces (surfaces with "pinched" points).

Let G be a graph with a cycle double cover $\mathcal{D}$. Two vertices or edges of G are said to be *confluent* if there is a disk containing both of them. If $\mathcal{D}$ is a cycle double cover in a graph G and S is a subdivision of G, then $\mathcal{D}$ induces a cycle double cover $\mathcal{D}'$ in S in the obvious way, and vice versa. We say that $\mathcal{D}'$ is the *disk system induced in S by $\mathcal{D}$*.

Let v be a vertex of a graph G with degree at least 4. Partition the set of its neighbors into two disjoint sets N_1 and N_2, with at least two vertices in each set. Let G' be obtained from G by replacing the vertex v with two adjacent vertices v_1, v_2, with v_i adjacent to the vertices in N_i for $i = 1, 2$. The graph G' is said to be obtained from G by *splitting* the vertex v. It is easy to see that if G is 3-connected, then so is G'. The vertices v_1 and v_2 are called the *new vertices* of G' and the edge v_1v_2 of G' is called the *new edge* of G'.

Suppose a graph G has a cycle double cover $\mathcal{D}$. The above splitting operation on a vertex v of G is said to be a *conforming split* (with respect to $\mathcal{D}$) if

(S1) among the disks that use the vertex v, there are exactly two, say D_1 and D_2, that use one vertex each from N_1 and N_2, and

(S2) D_1 and D_2 intersect precisely in the vertex v.

The split is then said to be *along* D_1 (and along D_2). A split that is not conforming as above is said to be a *non-conforming split*.

Let G, G' be as in the above paragraph. If G is a 3-connected planar graph, then G' is planar if and only if the split is conforming with respect to the disk system of peripheral cycles of G. More generally, to each cycle C of G there corresponds a unique cycle C' of G', and so to $\mathcal{D}$ there corresponds a uniquely defined set of cycles $\mathcal{D}'$ of G'. If $\mathcal{D}$ is a weak disk system, then so is $\mathcal{D}'$, and if $\mathcal{D}$ is a disk system, then so is $\mathcal{D}'$. We call $\mathcal{D}'$ the (weak) disk system *induced* in G' by $\mathcal{D}$. This is the purpose of conditions (S1) and (S2). If $\mathcal{D}$ is a disk system, then an equivalent way to define a conforming split of a vertex v is to say that both N_1 and N_2 form contiguous intervals in the cyclic order induced on the neighborhood of v by $\mathcal{D}$. Similarly, an equivalent condition for a split to be non-conforming with respect to a disk system is the existence of vertices $a, c \in N_1$ and $b, d \in N_2$ such that a, b, c and d appear in the cyclic order listed around v (as given by $\mathcal{D}$ in (D2)). The reason we use the definition above is that it applies more generally to weak disk systems.

A graph G' obtained from a graph G by repeatedly splitting vertices of degree at least four is said to be an *expansion* of G. In particular, each graph is an expansion of itself. Each split leading to an expansion of G has exactly one new edge; the set of these edges are the *new edges of the expansion of* G. The new edges form a forest in G'. If G has a cycle double cover $\mathcal{D}$, the expansion is called a *conforming* expansion if each of the splits involved in it is conforming (with respect to $\mathcal{D}$). If at least one of the splits involved is not conforming, then the expansion is called *non-conforming.* From the above discussion, it is clear that a disk system in G induces a unique disk system in a conforming expansion.

We now describe seven enlargement operations. Let G be a graph with a cycle double cover $\mathcal{D}$, and let G^+ be the graph obtained from G by applying one of the operation described below.

1. (non-conforming jump) G^+ is obtained from G by adding an edge uv where u and v are non-confluent vertices of G.

2. (cross) Let a, b, c, d be vertices appearing on a disk of G in that cyclic order. Add the edges ac and bd to obtain G^+.

3. (non-conforming split) G^+ is obtained from G by performing a non-conforming split of a vertex of G.

4. (split + non-conforming jump) Let u, v be non-adjacent vertices on some disk $C \in \mathcal{D}$. Perform a conforming split of v into v_1, v_2 such that u and v_2 are non-confluent vertices. (In particular, the split is not along C.) Now add the edge uv_2 to obtain G^+.

5. (double split + non-conforming jump) Let u, v be adjacent vertices and C_1, C_2 be the two disks containing the edge uv. Make a conforming split of u into u_1, u_2 along C_1 and a conforming split of v into v_1, v_2 along C_2 such that both splits are conforming and u_1 and v_1 are adjacent in the resulting graph. Now add the edge u_2v_2 to obtain G^+.

6. (split + cross) Let u, v, w be vertices on a disk C such that u is not adjacent to v or w. Perform a conforming split of u into u_1, u_2, along C, with u_1, u_2, v, w in that cyclic order on the new disk corresponding to C. Now add the edges u_1v and u_2w to obtain G^+.

7. (double split + cross) Let u, v be non-adjacent vertices on a disk C. Perform conforming splits of u and v, into u_1, u_2 and v_1, v_2, respectively such that both splits are along C. Let u_1, u_2, v_1, v_2 appear in that cyclic order on the new disk corresponding to C. Now add the edges u_1v_1 and u_2v_2 to obtain G^+.

If G^+ is obtained as in paragraph i above, then we say that G^+ is an *i-enlargement of G with respect to $\mathcal{D}$*. When the disk system $\mathcal{D}$ is implied by context, we may simply refer to an *i-enlargement* of G. We are now ready to state a preliminary form of our main result, a counterpart of Theorem 2.1.1, with minors instead of topological minors.

Theorem 2.1.2 *Let G be a weakly 4-connected planar graph, let H be a weakly 4-connected non-planar graph such that G is isomorphic to a minor of H, and let $\mathcal{D}$ be the disk system in G consisting of all peripheral cycles. Then there exists an integer $i \in \{1, 2, \ldots, 7\}$ such that H has a minor isomorphic to an i-enlargement of G with respect to $\mathcal{D}$.*

Theorem 2.1.1 is definitely easier to state than Theorem 2.1.2. So what are the advantages of the latter result? First, in the applications one is usually concerned with minors rather than topological minors, and so Theorem 2.1.2 gives a more direct route to the desired results. Second, while the number of types of outcome is larger in Theorem 2.1.2, in most cases the actual number of cases needed to examine will be smaller. (Notice that, for instance, in Theorem 2.1.1 one must examine all S-paths between non-confluent ends, whereas in Theorem 2.1.2 one is only concerned with those between non-confluent branch-vertices.)

Third, while a graph listed as an outcome of Theorem 2.1.1 may fail to be weakly 4-connected (and may do so in a substantial way), an i-enlargement of a weakly 4-connected graph is again weakly 4-connected. This has two advantages. In the applications we are often seeking to prove that weakly 4-connected graphs, with a minor isomorphic to some weakly 4-connected graph embeddable in a surface Σ, that themselves do not embed into Σ have a minor isomorphic to a member of a specified list $\mathcal{L}$ of graphs. In order to get a meaningful result we would like each member of $\mathcal{L}$ to satisfy the same connectivity requirement imposed on the input graphs.

From a more practical viewpoint, the advantage of maintaining the same connectivity in the outcome graph is that the theorem can then be applied repeatedly. That will become important when we consider a generalization to arbitrary surfaces (that is, in the context of theorems 2.2.1 and 2.7.5). While a weakly 4-connected graph G has at most one planar embedding, it may have several embeddings in a non-planar surface Σ. Now one application of the generalization of Theorem 2.1.2 will dispose of one embedding into Σ, but some other embedding might extend naturally to those outcome graphs. So it may be necessary to apply the theorem in turn to those outcome graphs in place of G. It will be important that the outcomes of (the generalization of) Theorem 2.1.2 satisfy the same requirement as the input graph. We can then apply such a theorem repeatedly till we get a list of graphs that no longer embed in Σ — in other words, we would have obtained the non-embeddable extensions of G. This will be illustrated in Section 2.8.

2.2 Main Theorem

Our main theorem applies to arbitrary disks systems, at the expense of having to add two additional outcomes, the following. As before, let G be a graph with a cycle double cover $\mathcal{D}$, and let G^+ be obtained by one of the operations below.

8. (non-separating triad) Let x_1, x_2, x_3 be three vertices of G such that (i) they are pairwise confluent, but not all contained in any single disk, and (ii) $\{x_1, x_2, x_3\}$ is independent, and does not separate G. To obtain G^+, add a new vertex to G adjacent to x_1, x_2 and x_3.

9. (non-conforming T-edge) Let a vertex u and an edge xy be such that (i) u is not confluent with the edge xy, but is confluent with both x and y, (ii) u is not adjacent to either x or y, and (iii) $\{u, x, y\}$ does not separate G. Subdivide the edge xy and join u to the new vertex, to obtain G^+.

As before, if G^+ is obtained as in paragraph i above, then we say that G^+ is an *i-enlargement of G with respect to $\mathcal{D}$.*

We also need to define an appropriate analogue of being non-planar in the context of cycle double covers. That is the objective of this paragraph and the next. Let S be a subgraph of a graph H. An *S-bridge of H* is a subgraph B of H such that either B consists of a unique edge of $E(H) - E(S)$ and its ends, where the ends belong to S, or B consists of a component J of $H \backslash V(S)$ together with all edges from $V(J)$ to $V(S)$ and all their ends. For an S-bridge B, the vertices of $B \cap S$ are called the *attachments* of B. Let $\mathcal{D}$ be a cycle double cover in S. We say that $\mathcal{D}$ is *locally planar* in H if the following conditions are satisfied:

(i) for every S-bridge B of H there exists a disk $C_B \in \mathcal{D}$ such that all the attachments of B lie on C_B, and

(ii) for every disk $C \in \mathcal{D}$ the subgraph $\bigcup B \cup C$ of H has a planar drawing with C bounding the unbounded face, where the big union is taken over all S-bridges B of H with $C_B = C$.

Let G have a weak disk system $\mathcal{D}$ and H have a minor isomorphic to G. It is easy to see that there is an expansion G' of G, such that G' is a topological minor of H. We say that $\mathcal{D}$ has a *locally planar extension* into H if:

(i) there exists a *conforming* expansion G' of G such that a subdivision of G' is a isomorphic to a subgraph S of H, and

(ii) the weak disk system $\mathcal{D}'$ induced in S by $\mathcal{D}$ is locally planar in H.

We are now ready to state the main result. A graph is a *prism* if its complement is a cycle of length six.

Theorem 2.2.1 *Let G and H be weakly 4-connected graphs such that H has a minor isomorphic to G and G is not a prism. Let G have a disk system $\mathcal{D}$ that has no locally planar extension into H. Then H has a minor isomorphic to an i-enlargement of G, for some $i \in \{1, \ldots, 9\}$.*

Let us deduce Theorem 2.1.2 from Theorem 2.2.1.

Proof of Theorem 2.1.2, assuming Theorem 2.2.1. Let $i \in \{8, 9\}$. By Theorem 2.2.1 it suffices to show that a weakly 4-connected planar graph G has no i-enlargement with respect to the disk system $\mathcal{D}$ consisting of all peripheral cycles of G. Indeed, suppose for a contradiction that such an i-enlargement exists, and let u, x, y be the three vertices of G as in the definition of i-enlargement. Since every pair of vertices among u, x, y are confluent, it follows that $G\backslash\{u, x, y\}$ is disconnected, a contradiction. □

2.3 Outline of Proof

The purpose of this section is to outline the proof of the main theorem. In fact, we prove a more general result, stated as Theorem 2.7.5, that applies to weak disk systems. However, this more general theorem requires three additional outcomes, and we introduce them now. Let J be a K_4 subgraph of a graph G, and assume that each of the four peripheral cycles of J are disks of G. In those circumstances we say that J is a *detached K_4 subgraph* of G. Again, let G be a graph with a cycle double cover $\mathcal{D}$, and let G^+ be obtained from G by one of the operations below.

10. (non-conforming double split) Let v be a vertex of G of degree at least 6. Partition the set of edges incident with v into three disjoint sets E_1, E_2, E_3, each of size at least 2, such that of the disks that use the vertex v, exactly two, say D_1 and D_2, use edges of two distinct sets E_i. Moreover, let D_1 and D_2 intersect precisely in the vertex v and let them use no edge of E_3. Then G^+ is obtained from G by replacing v with three vertices v_1, v_2, v_3, where v_i is incident with edges in E_i (for $i = 1, 2, 3$) and $G^+[\{v_1, v_2, v_3\}]$ has edge-set $\{v_1v_3, v_2v_3\}$.

11. (simplex on detached K_4) The graph G^+ is obtained from G by adding a new vertex adjacent to all four vertices of a detached K_4 subgraph of G.

12. (enlargement of a prism) Let G be a prism, and let G^+ be obtained from G by selecting two edges of G that do not belong to a common peripheral cycle but both belong to a triangle, subdividing them, and joining the two new vertices by an edge.

Similarly as before, if G^+ is obtained as in paragraph i above, then we say that G^+ is an *i-enlargement of G with respect to $\mathcal{D}$*. Thus if G is not a prism, then it has no 12-enlargement, and if G is a prism, then its 12-enlargement is unique, up to isomorphism. The unique 12-enlargement of the prism is known as V_8.

Our main tool for the proof of Theorem 2.2.1 will be its counterpart for subdivisions, proved in [41]. Before we can state it we need one more definition. Let S be a subgraph of a graph H, and let $\mathcal{D}$ be a cycle double cover in S. Let $x \in V(H) - V(S)$ and let x_1, x_2, x_3 be distinct vertices of S such that every two of them are confluent, but no disk of S contains all three. Let L_1, L_2, L_3 be three paths such that (i) they share a common end x, (ii) they share no internal vertex among themselves or with S, and (iii) the other end of L_i is x_i, for $i = 1, 2, 3$. The paths L_1, L_2, L_3 are then said to form an *S-triad*. The vertices x_1, x_2, x_3 are called the *feet* of the triad. We are now ready to state the result from [41].

Theorem 2.3.1 ([41]) *Let G be a graph with no vertices of degree two, let H be a weakly 4-connected graph, let $\mathcal{D}$ be a weak disk system in G, and let a subdivision of G be isomorphic to a subgraph of H. Then there exists a subgraph S of H isomorphic to a subdivision of G such that, letting $\mathcal{D}'$ denote the weak disk system induced in S by $\mathcal{D}$, one of the following conditions holds:*

1. *there exists an S-path in H such that its ends are not confluent in S, or*
2. *there exists a free S-cross in H on some disk of S, or*
3. *the graph H has an S-triad, or*
4. *the graph S has a detached K_4 subgraph J such that the attachments of some S-bridge of H are precisely the branch-vertices of J, or*
5. *the weak disk system $\mathcal{D}'$ is locally planar in H.*

Now let G, $\mathcal{D}$ and H be as in Theorem 2.2.1. It is easy to see that there exists an expansion G' of G such that a subdivision of G' is isomorphic to a subgraph S of H. (If G itself is a topological minor of H, then $G' = G$.) In Lemma 2.4.4 we prove that if G' is a nonconforming expansion, then there exists a 3-enlargement or a 10-enlargement of G that is isomorphic to a minor of H. Thus from now on we may assume that G' is a conforming expansion of G. By Lemma 2.3.1 applied to S and H we deduce that one of the outcomes of that lemma holds. Notice that those outcomes correspond to 1-enlargement, 2-enlargement, 8-enlargement and 11-enlargement, respectively, except that in the enlargements the vertices in question are required to be branch-vertices of S, whereas in Lemma 2.3.1 they are allowed to be interior vertices of segments. We deal with this in Section 2.5 by showing that each of the outcomes mentioned leads to a suitable enlargement of G'. To be precise, at this point we settle for what we call weak 8- and weak 9-enlargements, and in Section 2.6 show that these weak enlargements can be replaced by ordinary enlargements, possibly of a different expansion of G and of a different kind. Finally, in Section 2.7 we complete the proof of Theorem 2.2.1 by showing that the expansion G' can be chosen to be equal to G.

2.4 Preliminaries

Let G' be an expansion of a graph G. Then every vertex v of G corresponds to a connected subgraph T_v of G'. We call $V(T_v)$ the *branch-set corresponding to* v.

Lemma 2.4.1 *Let G' be an expansion of a graph G, let $u, v \in V(G)$ be distinct, and let T_u, T_v be the corresponding subgraphs of G'. Then T_u and T_v are induced subtrees of G'. If u is adjacent to v then exactly one edge of G' has one end in $V(T_u)$ and the other in $V(T_v)$, and if u is not adjacent to v, then no such edge exists.*

An expansion of a weakly 4-connected graph may fail to be weakly 4-connected, but only in a limited way. The next definition and lemma make that precise. Let (A, B) be a nontrivial separation of order three in a graph G. We say that (A, B) is *degenerate* if the vertices in $A \cap B$ can be numbered v_1, v_2, v_3 such that either

(1) $|A - B| = 1$ and $A \cap B$ is an independent set, or

(2) there exists a triangle $u_1u_2u_3$ in $G[A]$ such that for $i = 1, 2, 3$ the vertices u_i and v_i are either adjacent or equal, $A \subseteq \{u_1, u_2, u_3, v_1, v_2, v_3\}$, and each edge of $G[A]$ is of the form u_iv_i for $1 \le i \le 3$ or u_iu_j for $1 \le i < j \le 3$.

The following two lemmas are routine, and we omit the straightforward proof.

Lemma 2.4.2 *Let G be an expansion of a weakly 4-connected graph. Then G is 3-connected, and if it is not a prism, then for every nontrivial separation (A, B) of G of order three, exactly one of (A, B), (B, A) is degenerate.*

Lemma 2.4.3 *Let G' be expansion of a weakly 4-connected graph G, let (A, B) be a degenerate separation of G of order three satisfying condition (2) of the definition of degenerate separation, and let $u_1, u_2, u_3, v_1, v_2, v_3$ be as in that condition. Then for at least two integers $i \in \{1, 2, 3\}$ either $u_i = v_i$ or $u_i v_i$ is a new edge of G'.*

We now show that a non-conforming expansion of G must have a minor isomorphic to a 3-enlargement or a 10-enlargement of G.

Lemma 2.4.4 *Let G' be a non-conforming expansion of a graph G. Then G' has a minor isomorphic to a 3-enlargement or 10-enlargement of G.*

Proof: We may assume that for every new edge e of G' the graph G'/e is a conforming expansion of G. We shall refer to this as the *minimality* of G'. We will prove that G' is either a 3-enlargement or a 10-enlargement of G.

Let $\widehat{G}$ be an expansion of G such that G' is obtained from $\widehat{G}$ by splitting a vertex v into v_1 and v_2. By the minimality of G' this split is non-conforming, and $\widehat{G}$ is a conforming expansion of G. If $G = \widehat{G}$, then G' is a 3-enlargement of G, and so we may assume that $G \neq \widehat{G}$. Let e be a new edge of $\widehat{G}$. If e is not incident with v, then G'/e is a non-conforming expansion of $\widehat{G}/e$, contrary to the minimality of G'. Now let us consider e as an edge of G'. From the symmetry between v_1 and v_2 we may assume that e is incident with v_2 in G'; let v_3 be its other end. The split of the vertex v of the graph $\widehat{G}$ into v_1 and v_2 violates (S1) or (S2). But it does not violate (S1), for otherwise the same violation occurs in the analogous split of $\widehat{G}/e$, contrary to the minimality of G'. Thus the split of the vertex v of the graph $\widehat{G}$ into v_1 and v_2 satisfies (S1); let D_1 and D_2 be the corresponding disks. It follows that the disks violate (S2), but they do not do so for the corresponding split in $\widehat{G}/e$. It follows that $e \in E(D_1) \cap E(D_2)$. Since G'/e is a conforming expansion of G it follows that G' is a 10-enlargement of $\widehat{G}/e$. But $\widehat{G}/e = G$ by the minimality of G, as desired. □

The following lemma will be useful.

Lemma 2.4.5 *Let G' be a conforming expansion of a graph G with respect to a weak disk system $\mathcal{D}$, and let $\mathcal{D}'$ be the weak disk system induced in G' by $\mathcal{D}$. Let qr be a new edge of G', and let the vertex $p \in V(G') - \{q, r\}$ share distinct disks D_q, D_r of G' with q and r, respectively, such that D_r does not contain q. Then p is adjacent to r and the disks D_q, D_r both contain the edge pr.*

Proof: The disks of G/qr that correspond to D_q and D_r share p and the new vertex of G/qr, say w. By (D3) p is adjacent to w in G/qr and the edge pw belongs to both those disks. By Lemma 2.4.1 the vertex p is adjacent to exactly one of q, r. But $q \notin V(D_r)$, and hence p is adjacent to r and D_q, D_r both contain the edge pr, as desired. □

We end this section with a lemma about fixing separations in weakly 4-connected graphs, a special case of a lemma from [23]. First some additional notation: when a graph G is a minor of a graph H, we say that an *embedding* η of G into H is a mapping with domain $V(G) \cup E(G)$ as follows. η maps vertices $v \in G$ to connected subgraphs $\eta(v)$ of H, with distinct vertices being mapped to disjoint vertex-disjoint subgraphs. Further, η maps edges uv of G to paths $\eta(uv)$ in H with one end in $\eta(u)$ and the other in $\eta(v)$, and otherwise disjoint from $\eta(w)$ for any vertex w of G. Also, for edges $e \neq e'$ of G, if $\eta(e)$ and $\eta(e')$ share a vertex, then it must be an end of both the paths.

Lemma 2.4.6 *Let G_1 be a graph isomorphic to a minor of a weakly 4-connected graph H. Let $P = \{p_1, p_2\}$, $Q = \{q_1, q_2, q_3\}$ and R be such that (P, Q, R) is a partition of $V(G_1)$, and G_1 has all possible edges between P and Q, and no edge with both ends in Q. Further, suppose R has at least two vertices, and that $(P \cup Q, Q \cup R)$ is a (non-trivial) 3-separation of G_1. Then H has a minor isomorphic to a graph G_1^+ that is obtained from G_1 by*

1. *adding an edge between p_i and r for some $i \in \{1, 2\}$ and $r \in R$, or*

2. *splitting q_j for some $j \in \{1, 2, 3\}$ into vertices q_j^1 and q_j^2 such that q_j^1 is adjacent to p_1 and q_j^2 is adjacent to p_2*

Proof: Call an embedding η of G_1 into H *minimal* if for every embedding η' of G into H,

$$\sum_{j=1}^{3} |E(\eta(q_j))| \leq \sum_{j=1}^{3} |E(\eta'(q_j))|$$

In particular, if η is minimal, $\eta(q_j)$ is a tree for every j. Further, we say that a vertex q_j is *good* for η if the paths $\eta(p_1q_j)$ and $\eta(p_2q_j)$ are vertex-disjoint (in other words, their ends in $\eta(q_j)$ are distinct).

Consider a minimal embedding η of G_1 into H. Suppose there exists a q_j that is good for η. For $i = 1, 2$, let p_i' be the endpoint of $\eta(p_iq_j)$ in $\eta(q_j)$. Let e be an edge in the unique path between p_1' and p_2' in $\eta(q_j)$, and let T_1, T_2 be the two subtrees obtained by deleting e from $\eta(q_j)$, such that $p_i' \in T_i$ for $i = 1, 2$. For $i = 1, 2$, define N_i to be the set of neighbors $r \in R$ of q_j in G that $\eta(rq_j)$ has an endpoint in T_i. Now N_1, N_2 are non-empty by the minimality of η. (If, say, N_1 were empty, then we could replace $\eta(q_j)$ by T_2 and modify $\eta(p_1q_j)$ accordingly to get a better embedding η', a contradiction.) It is easy to see that conclusion 2 of the lemma is satisfied, with the neighborhoods of q_j^1 and q_j^2 being $N_1 \cup \{p_1\}$ and $N_2 \cup \{p_2\}$, respectively.

Hence we may assume that there is no minimal embedding of G into H with a vertex in Q being good for it. Let η be an embedding of G into H. For $j = 1, 2, 3$, there exist vertices t_j such that both $\eta(p_1q_j)$ and $\eta(p_2q_j)$ have t_j as an end. Define J_1 as the union of $\eta(p_i)$, $i = 1, 2$ and of $\eta(e)$ for all edges e with at least one end in P. Define J_2 as the union of $\eta(v)$ for $v \in Q \cup R$ and of $\eta(e)$ for every edge e of G with both ends in $Q \cup R$. Now $V(J_1) \cap V(J_2) = \{t_1, t_2, t_3\}$. Since H is weakly 4-connected, there is a path in H with ends $a \in V(J_1) \setminus V(J_2)$ and $b \in V(J_2) \setminus V(J_1)$, and otherwise disjoint from $J_1 \cup J_2$. If b belongs to $\eta(q_j) \setminus t_j$ for some j, then we can modify η to get a minimal embedding where q_j is a good vertex, which is a contradiction. Thus b belongs to $\eta(r)$ for some $r \in R$ or b is an internal vertex of $\eta(e)$ for an edge e of G that has an end in R (recall that Q is an independent set). In either case, it is easy to see that conclusion 1 holds. □

2.5 The Enlargements of an Expansion of G

Let G and H be as in Theorem 2.2.1. In order to apply Theorem 2.3.1 we select an expansion G' of G such that a subdivision of G' is isomorphic to a subgraph of H. By Lemma 2.4.4 we may assume that G' is a conforming expansion. In this section we prove three lemmas, one corresponding to each of the first three outcomes of Theorem 2.3.1. The lemmas together almost imply that the conclusion of Theorem 2.2.1 holds for G'. The reason for the word almost is that for convenience we allow a weaker form of 8-enlargements and 9-enlargements.

The weaker form of 9-enlargements is defined as follows. Let G be a graph with a cycle double cover $\mathcal{D}$, and let $u, x, y \in V(G)$, where x and y are adjacent and u is not confluent with the edge xy. Let G^+ be obtained from G by subdividing the edge xy and adding an edge joining the new vertex to u. We say that G^+ is a weak 9-enlargement of G. Later, in Lemma 2.6.3, we show how to move from a weak 9-enlargement to a 9-enlargement or another useful outcome. Our first lemma deals with the first outcome of Theorem 2.3.1.

Lemma 2.5.1 *Let G, H be graphs such that G is connected, has at least five vertices and no vertices of degree two. Let $\mathcal{D}$ be a weak disk system in G, let S be a subgraph of H isomorphic to a subdivision of G, and let P be an S-path in H such that its ends are not confluent in the weak disk system $\mathcal{D}'$ induced in S by $\mathcal{D}$. Then H has a minor isomorphic to a 1-enlargement, 3-enlargement or a weak 9-enlargement of G.*

Proof: Let s, t be the ends of P. If both s and t are branch-vertices in S, then H has a minor isomorphic to a 1-enlargement of G, and we are done. If one of s and t is a branch-vertex and the other is an internal vertex of a segment of S, then H has a minor isomorphic to a weak 9-enlargement of G, as desired.

Thus we may assume that s and t are internal vertices of two different segments Q_1 and Q_2 of S, respectively. Let Q_1 correspond to an edge $uv \in E(G)$, and let Q_2 correspond to an edge $xy \in E(G)$. Now, if u is not confluent with the edge xy, then H has a minor isomorphic to a weak 9-enlargement of G, and the lemma holds. Thus, we may assume that u shares a disk D_1 with the edge xy. By symmetry, we get a disk D_2 shared by v and the edge xy, and disks D_3, D_4 that the edge uv shares with vertices x and y respectively. (The disks D_i may not be pairwise distinct.)

The disks D_1 and D_3, however, must be distinct, since the vertices s, t are not confluent. Notice, however, that they share the vertices u and x. It follows that u, v, x, y are pairwise distinct, for if $v = y$, say, then $u, v = y, x$ all belong to $V(D_1 \cap D_3)$, and hence $D_1 = D_3$ by (D3), a contradiction. By (D3) this implies that u is adjacent to x in G and the intersection of D_1 and D_3 is precisely the edge ux. In other words, the vertices u and x must be adjacent in G, and D_1, D_3 are precisely the two disks containing the edge ux. By a similar argument, it follows that u and y are adjacent, and D_1, D_4 are precisely the two disks containing the edge uy. Thus u is adjacent with each of v, x, y in G, and the edges uv, ux, uy are pairwise confluent.

By symmetry, we get similar conclusions about the vertices v, x, y. Thus $G[u, v, x, y]$ is a detached K_4 subgraph of G. Since G has at least five vertices and is connected, we may assume, without loss of generality, that u has a neighbor in G outside of the set $\{v, x, y\}$. Let N be the set of all such neighbors of u. But then delete the edges of the segment of S corresponding to the edge ux and contract the edges of the subpath of Q_2 between t and the end corresponding to x. It follows that H has a minor isomorphic to a graph obtained from G by splitting u corresponding to the partition $\{\{v, x\}, N \cup \{y\}\}$ of its neighbors. This split is non-conforming since the disks D_1 and D_4 violate condition (S2) in the definition of a conforming split. Hence H has a minor isomorphic to a 3-enlargement of G. $\square$

Lemma 2.5.2 *Let G, H be graphs such that H is weakly 4-connected, and G is connected, has at least 5 vertices and has no vertices of degree two. Let $\mathcal{D}$ be a weak disk system in G, let S be a subgraph of H isomorphic to a subdivision of G, such that $\mathcal{D}$ induces the weak disk system $\mathcal{D}'$ in S. Further, let there exist a free S-cross on some disk of S. Then H has a minor isomorphic to a 2-enlargement or a 3-enlargement or a weak 9-enlargement of G.*

Proof: Let the free cross consist of paths P_1, P_2, in a disk C' of S, that corresponds to a disk C of G. We shall call the paths P_1, P_2 the *legs* of the cross. Recall that the ends of P_1, P_2 are called the *feet* of the cross.

If C has at least four vertices, then we claim that H has a minor isomorphic to a 2-enlargement of G. We define an auxiliary bipartite graph B, with the vertex set being the set of feet of the cross and the set of branch-vertices of S that belong to C'. A foot f and a branch-vertex b are adjacent if one of the subpaths of C' with ends f and b includes no feet or branch-vertices in its interior. Since the cross is free, it follows from Hall's bipartite matching theorem that B has a complete matching from the set of feet to the set of branch vertices (in other words, one that matches each of the feet). By contracting the edges of the paths that correspond to this matching, we deduce that H has a minor isomorphic to a 2-enlargement of G, as desired.

Hence we may assume that C is in fact a triangle on vertices u_1, u_2 and u_3, say. For $i = 1, 2, 3$, if u_i has degree 3 in G, then define v_i to be its third neighbor (that is, the neighbor not in C). Otherwise, define $v_i = u_i$. Let $u'_1, u'_2, u'_3, v'_1, v'_2, v'_3$ be the corresponding vertices of S. Let Q_i denote the segment of S corresponding to the edge u_iv_i if $u_i \neq v_i$ and let Q_i be the null graph otherwise, and let $A = V(C' \cup P_1 \cup P_2)$ and $B = (V(S) - V(C' \cup Q_1 \cup Q_2 \cup Q_3)) \cup \{v'_1, v'_2, v'_3\}$. There exist three vertex-disjoint paths in H linking $\{u'_1, u'_2, u'_3\}$ to $\{v'_1, v'_2, v'_3\}$. Since H is weakly 4-connected, it follows that there is no 3-cut in H separating A from B. Hence, by a variant of Menger's theorem, H contains four vertex-disjoint paths $L_1, \dots L_4$ linking $\{v'_1, v'_2, v'_3, y\}$ to $\{u'_1, u'_2, u'_3, x\}$ (not necessarily in that order), where $x \in A$ and $y \in B$. We assume the numbering of the paths is such that for $i = 1, 2$ and 3, L_i has end $v'_i \in B$. (The remaining path L_4 then has end $y \in B$.) Let w be a nearest node of y that is not in $\{v_1, v_2, v_3\}$. (Note that such a nearest node exists by Lemma 2.4.1, and Lemma 2.4.3.)

We may assume that $x \in V(C')$. If not, then we may contract edges suitably in P_1 or P_2 such that the vertex corresponding to x, after the contraction, lies on C'. (Note that this contraction does not affect the graph S, neither does it destroy the cross.)

Relabel the vertices u'_1, u'_2, u'_3, x as a, b, c, d, in the order in which they appear on C' (in some orientation), such that L_4 joins d to y. (Note that d need not be the same as x.) Let (d, a, b) denote the interior vertices of the subpath of C' with ends d and b that includes a in its interior, and let (d, c, b) be defined analogously.

We claim that there is a leg of the cross with feet f, g such that $f \in (d, a, b)$ and $g \in (d, c, b)$. Since the cross is free, there exists a leg with foot in (d, a, b). We may assume the other foot of this leg does not belong to (d, c, b), but then the other leg of the cross satisfies the claim.

Choose a leg as above such that there is no foot between f and a, and no foot between g and c. (Such a choice must be possible, due to the freeness of the cross.) Let the other leg of the cross have feet h, i, such that b and h are joined by a subpath of the cycle C' that is disjoint from $\{f, g\}$. By contracting disjoint subpaths of C' with ends (a, f), (c, g), and (b, h) respectively, it follows that H has a minor isomorphic to the graph G' obtained from G by adding a new vertex z adjacent to u_1, u_2, u_3 and w.

If w is not confluent with the edge u_1u_2 then $G' \setminus u_1u_2 \setminus zu_3$ is isomorphic to a weak 9-enlargement of G, and we are done. Thus we may assume that w is confluent with the edge u_1u_2, and by symmetry, with the edges u_2u_3 and u_1u_3 as well. It follows similarly as in the proof of Lemma 2.5.1 that $G[u_1, u_2, u_3, w]$ is a detached K_4 subgraph of G. Since G is connected, and $|V(G)| \geq 5$, we may assume, without loss of generality, that u_1 has a neighbor in G outside that set. It follows that a graph obtained from G by a non-conforming split of u_1 is isomorphic to a minor of H. $\square$

We now define the weaker form of 8-enlargements. Let G be a graph with a cycle double cover $\mathcal{D}$, and let x_1, x_2, x_3 be vertices of G such that no disks contains all three. Let G^+ be obtained from G by adding a vertex with neighborhood $\{x_1, x_2, x_3\}$. We say that G^+ is a weak 8-enlargement of G. Our third lemma deals with the third outcome of Theorem 2.3.1.

Lemma 2.5.3 *Let G, H be graphs such that G is connected, has at least five vertices and no vertices of degree two. Let $\mathcal{D}$ be a weak disk system in G, let S be a subgraph of H isomorphic to a subdivision of G, and let there exist an S-triad in H. Then H has a minor isomorphic to an i-enlargement of G for $i = 1$ or 3, or a weak i-enlargement of G for $i = 8$ or 9.*

Proof: We proceed by induction of $|E(H)|$. Let the S-triad be L_1, L_2, L_3, and let its feet be x_1, x_2, x_3. If each x_i is a branch-vertex of S, then $S \cup L_1 \cup L_2 \cup L_3$ gives rise to a minor of H isomorphic to a weak 8-enlargement, as desired. We may therefore assume that x_3 is an internal vertex of a segment Q_3 of S with ends u_3 and v_3. Let f be an edge of Q_3. By induction applied to G, H/f, and S/f, we may assume that f is incident with x_3 and one end of Q_3, say u_3, and that there exists a disk D_1 in G containing x_1, x_2, u_3. Similarly, we may assume that there exists a disk D_2 in G containing x_1, x_2, v_3. Then $D_1 \neq D_2$, because otherwise $D_1 = D_2$ includes the segment Q_3 by (D3), and hence each of x_1, x_2, x_3, a contradiction. Since x_1 and x_2 belong to $D_1 \cap D_2$, it follows from (D3) that x_1 and x_2 belong to a common segment Q of S.

Let S' be obtained from S by replacing $Q[x_1, x_2]$ by $L_1 \cup L_2$. Applying Lemma 2.5.1 to G, H, S' and the S'-path L_3, the lemma now follows. □

Using Theorem 2.3.1 we can summarize Lemmas 2.5.1–2.5.3 as follows.

Lemma 2.5.4 *Let G, H be weakly 4-connected graphs, let G have a weak disk system $\mathcal{D}$ with no locally planar extension into H, and let a subdivision of G be isomorphic to a subgraph of H. Then H has a minor isomorphic to*

(i) an i-enlargement of G for some $i \in \{1, 2, 3, 11\}$, or

(ii) a weak i-enlargement of G for some $i \in \{8, 9\}$.

Proof: Let G, H and $\mathcal{D}$ be as stated. By Theorem 2.3.1 we deduce that there exists a subgraph S of H isomorphic to a subdivision of G such that the induced weak disk system in S satisfies one of the outcomes (1)–(4) of Theorem 2.3.1. If it satisfies (1)–(3), then (i) or (ii) of this lemma hold by Lemmas 2.5.1–2.5.3. If (4) of Theorem 2.3.1 holds, then H has a minor isomorphic to a 11-enlargement of G, as desired. □

2.6 From Weak Enlargements to Enlargements

The purpose of this section is to replace weak enlargements by enlargements in Lemma 2.5.4(ii). We start with a special case of weak 9-enlargements.

Lemma 2.6.1 *Let G be a graph with a cycle double cover $\mathcal{D}$, let G^+ be a weak 9-enlargement of G, and let u, x, y be as in the definition of weak 9-enlargement. If $G\backslash\{u, x, y\}$ is connected, then G^+ has a minor isomorphic to an i-enlargement of G for some $i \in \{1, 3, 9\}$.*

Proof: Let z be the new vertex of G^+ that resulted from the subdivision of the edge xy. If u and x are not confluent, then contracting the edge xz of G^+ produces a 1-enlargement of G, and so the lemma holds. Thus we may assume that u and x are confluent, and, by symmetry, we may assume that u and y are confluent. If u is not adjacent to x or y, then G^+ is a 9-enlargement of G, and the lemma holds. Thus, from the symmetry, we may assume that u is adjacent to x. The edges xy and xu are not confluent, for otherwise u is confluent with the edge xy, contrary to what a weak 9-enlargement stipulates. But then deleting the edge xu from G^+ yields a graph isomorphic to a 3-enlargement of G — more specifically, a graph obtained by a non-conforming split of the vertex x. □

Lemma 2.6.2 *Let G be an expansion of a weakly 4-connected graph, let H be a weakly 4-connected graph, let $\mathcal{D}$ be a weak disk system in G, let v be a vertex of G of degree three and let u, x, y be the neighbors of v. Let G^+ be the graph obtained from G by adding a new vertex z adjacent to u, x, y and deleting all edges with both ends in $\{u, x, y\}$. If H has a minor isomorphic to G^+, then H has a minor isomorphic to an i-enlargement of G for some $i \in \{1, 2, 3, 4, 9, 12\}$.*

Proof: Since G is an expansion of a weakly 4-connected graph, Lemma 2.4.2 implies that at most one edge of G has both ends in $\{u, x, y\}$. Thus we may assume that u is not adjacent to x or y. Since v has degree three, it follows from (D1) and (D3) that the triangle vxy is a disk in G. We can apply Lemma 2.4.6 to $G^+ = G_1$ and H, with $P = \{v, z\}$, $Q = \{u, x, y\}$ and $R = V(G^+) - (P \cup Q)$. From the lemma, using the symmetry between x and y, and the symmetry among x, y and u if x is not adjacent to y, we get the following three cases:

Case 1: H has a minor isomorphic to a graph G^{++} that is obtained from G^+ by adding an edge between a vertex $p \in P$ and a vertex $r \in R$. Note that the vertices v and z are symmetric for the application of Lemma 2.4.6. Hence we may assume that $p = v$. Now if

r is not confluent with v in G, then G^{++} above has a minor isomorphic to a 1-enlargement of G. Thus we may assume that r is confluent with v in G. Furthermore, we may assume, without loss of generality, that the disk D_3 shared by r and v contains the edges vu and vy. (Note that v has degree 3 in G.) On the disk D_3, the vertices u, v, y and r occur in that cyclic order. Now in G^{++}, contracting the edge yz gives a cross in the disk D_3 with arms uy and rv. In other words, G^{++}, and hence H, has a minor isomorphic to a 2-enlargement of G, as desired.

Case 2: The vertices x and y are adjacent in G and H has a minor isomorphic to a graph G^{++} that is obtained from G^{+} by splitting the vertex x into x_1 and x_2, with x_1 adjacent to v and x_2 adjacent to z. Let N_i be the neighbors of x_i in G^{++} other than v, z, x_1, x_2. The neighborhood of x in G is thus $N_1 \cup N_2 \cup \{v, y\}$. In G, let D_4 be the disk that contains the edge xy, other than the triangle vxy. The disk D_4 must contain a vertex in either N_1 or N_2, and from the symmetry between v and z we may assume that it contains a vertex in N_1. Then, in G^{++}, delete the edge uz and contract the edge x_2z. This gives a graph that is a 3-enlargement of G (non-conforming split of x, with the disks vxy and D_4 violating condition (S2) in the definition of a conforming split), as desired.

Case 3: H has a minor isomorphic to a graph G^{++} that is obtained from G^{+} by splitting the vertex u into u_1 and u_2, with u_1 adjacent to v and u_2 adjacent to z. Let N_i be the set of neighbors of u_i other than v, z, u_1, u_2. Thus in G, the neighborhood of u is $N_1 \cup N_2 \cup \{v\}$.

Let D_1 be the disk in G shared by the edges xv and vu, and D_2 be the disk in G shared by the edges yv and vu. The disks D_1 and D_2 both contain exactly one vertex each from $N_1 \cup N_2$. Let us assume first that $|N_2| \geq 2$. Contract the edge xz in G^{++}, and if x is not adjacent to y in G, then delete also the resulting edge xy to obtain a graph G_1, and let G_2 be the graph obtained from G_1 by further deleting the edge u_2x. Now G_2 is isomorphic to a graph obtained from G by splitting the vertex u into u_1 and u_2. If this split is non-conforming, then G_2 is a 3-enlargement of G, and we are done. Otherwise, the split is not along D_1 or D_2, and from the symmetry we may assume it is not along D_1. Thus G_1 is a 4-enlargement of G. (Note that in G, u and x are non-adjacent, and hence non-consecutive on D_1.) This completes the case when $|N_2| \geq 2$.

From the symmetry we may therefore assume that $|N_1| = |N_2| = 1$. Thus the degree of u in G is three. For $i = 1, 2$ let $N_i = \{n_i\}$. We may assume that the edge un_i belongs to the disk D_i. It follows that the vertex x and edge un_2 are not confluent in G, for if some disk D contained both of them, then the intersection $D \cap D_1$ would violate (D3), because u is not adjacent to x. The graph G_1 from the previous paragraph is a weak 9-enlargement of G, and so by Lemma 2.6.1 we may assume that $G \backslash \{x, u, n_2\}$ is disconnected. Since u has degree three, the weak 4-connectivity of G implies that n_1 has degree three and its neighbors are x, u, n_2. Since $G \backslash \{n_2, y\}$ is connected, we deduce that G is isomorphic to the prism, and G^{++} is isomorphic to a 12-enlargement of G, as desired. □

Now we are ready to eliminate weak 9-enlargements.

Lemma 2.6.3 *Let G be an expansion of a weakly 4-connected graph, let $\mathcal{D}$ be a weak disk system in G, and let G^+ be a weak 9-enlargement of G such that G^+ is isomorphic to a minor of a weakly 4-connected graph H. Then H has a minor isomorphic to an i-enlargement of G for some $i \in \{1, 2, 3, 4, 9, 12\}$.*

Proof: Let u, x, y be as in the definition of weak 9-enlargement. By Lemma 2.6.1 we may assume that $G \backslash \{u, x, y\}$ is disconnected. Since x is adjacent to y and G is an expansion of a weakly 4-connected graph, Lemma 2.4.2 implies that the neighborhood of some vertex v of G is precisely the set $\{u, x, y\}$. Thus G^+ is as described in Lemma 2.6.2, and the conclusion follows from that lemma. □

We now turn to weak 8-enlargements. In order to save effort we prove a weaker analogue of Lemma 2.6.3, the following.

Lemma 2.6.4 *Let G_1 be an expansion of a weakly 4-connected graph G, let $\mathcal{D}$ be a weak disk system in G, and let G^+ be a weak 8-enlargement of G_1 such that G^+ is isomorphic to a minor of a weakly 4-connected graph H. Then there exists an expansion G_2 of G obtained from G_1 by contracting a possibly empty set of new edges such that H has a minor isomorphic to an i-enlargement of G_2 for some $i \in \{1, 2, 3, 4, 8, 9, 12\}$.*

Proof: We proceed by induction on $|E(G_1)|$. Let G^+ be obtained from G_1 by adding a vertex joined to v_1, v_2, v_3. If some edge of G_1 has both ends in the set $\{v_1, v_2, v_3\}$, then by deleting that edge we obtain a graph isomorphic to a weak 9-enlargement of G_1, and the lemma follows from Lemma 2.6.3. Thus we may assume that $\{v_1, v_2, v_3\}$ is an independent set in G_1. We may also assume that every pair of vertices in $\{v_1, v_2, v_3\}$ is confluent, for otherwise G^+ has a minor isomorphic to a 1-enlargement of G, and the lemma holds. Thus we may assume that $G \backslash \{v_1, v_2, v_3\}$ is disconnected, for otherwise G^+ is an 8-enlargement of G_1.

Let (A, B) be a non-trivial separation of G with $A \cap B = \{v_1, v_2, v_3\}$. By Lemma 2.4.2 we may assume that (A, B) is degenerate. If $|A - B| = 1$, then the lemma follows from Lemma 2.6.2. Thus we may assume that $|A - B| \geq 2$. Let $v_1, v_2, v_3, u_1, u_2, u_3$ be as in the definition of degenerate. Since $\{v_1, v_2, v_3\}$ is independent, we may assume from the symmetry that $u_1 \neq v_1$ and $u_2 \neq v_2$. Now one of u_1v_1, u_2v_2 is a new edge of G_1, and so we may assume the former is. Thus G^+/u_1v_1 is a weak 8-enlargement of G_1/u_1v_1, and hence the lemma follows by the induction hypothesis applied to the graph G_1/u_1v_1. □

The lemmas of this section allow us to upgrade Lemma 2.5.4 to the following.

Lemma 2.6.5 *Let G, H be weakly 4-connected graphs, let G have a weak disk system $\mathcal{D}$ with no locally planar extension into H, and let G' be a conforming expansion of G such that a subdivision of G' is isomorphic to a subgraph of H. Then there exists a conforming expansion G'' of G obtained from G' by contracting a possibly empty set of new edges such that, letting $\mathcal{D}''$ denote the weak disk system induced in G'' by $\mathcal{D}$, the graph H has a minor isomorphic to an i-enlargement of G'' with respect to $\mathcal{D}''$ for some $i \in \{1, 2, 3, 4, 8, 9, 11, 12\}$.*

Proof: By Lemma 2.5.4 we may assume that a weak 8-enlargement or a weak 9-enlargement of G' is isomorphic to a minor of H. By Lemmas 2.6.3 and 2.6.4 there exists a required conforming expansion G'' of G such that H has a minor isomorphic to an i-enlargement of G'' for some $i \in \{1, 2, 3, 4, 8, 9, 12\}$. □

2.7 Proof of the Main Theorem

Lemma 2.6.5 gives an i-enlargement of an expansion G'' of G. Our final objective is to show that we can choose $G'' = G$. We break the proof into several lemmas depending on the value of i.

Lemma 2.7.1 *Let G and H be weakly 4-connected graphs, and let $\mathcal{D}$ be a weak disk system in G with no locally planar extension into H. Let G' be a conforming expansion of G such that H has a minor isomorphic to a 1-enlargement of G'. Then H has a minor isomorphic to an i-enlargement of G for some $i \in \{1, 3, 4, 5\}$.*

Proof: We may assume that G' is as stated in the lemma, and subject to that, it is minor-minimal. By hypothesis, H has a minor isomorphic to G^+, a graph obtained from G' by adding an edge between two vertices x and y that are not confluent. Let e be a new edge of G'. By the minimality of G', it follows that

(i) one end of e must be in $\{x, y\}$, and

(ii) the other end of e must be confluent with the vertex in $\{x, y\}$ other than the one above.

Recall that branch-sets of an expansion were defined at the beginning of Section 2.4. Thus all branch sets that are disjoint from $\{x, y\}$ are singleton sets. Let T_p and T_q be the branch sets corresponding to vertices $p, q \in V(G)$ such that they contain x and y respectively (p and q may be identical). We claim that the degree of x in the branch set containing it is at most one (that is, x is a leaf of the tree $G'[T_p]$). Suppose not; hence x has (at least) two neighbors x_1 and x_2 in T_p. By (ii) above, y shares disks D_1 and D_2 of G' with x_1 and x_2 respectively. Then $x \notin V(D_1 \cup D_2)$, for x, y are not confluent. It follows that $D_1 \neq D_2$, for otherwise D_1 is not a cycle in $G'/x_1x/x_2x$, and yet D_1 corresponds to a disk in G. Also, y is not adjacent to both x_1 and x_2, by Lemma 2.4.1. But then contracting edges xx_1 and xx_2 violates Axiom (D3) in G. This proves the claim. Thus x, and by symmetry y, are leaf vertices in $G'[T_p]$ and $G'[T_q]$ respectively.

If $p = q$, then it follows that $T_p = T_q$ must be a path of length 2, with a middle vertex z. Let $D'_1, \mathcal{D}'_2$ be the two disks in G' that include the edge xz, and let D'_3, D'_4 be the two disks that include the edge yz. Note that, since x and y are not confluent in G', all four disks above are distinct. Let D_1, D_2, D_3, D_4 be the corresponding disks in G. Let N_1, N_2 be the partition of the set of neighbors of p in G, corresponding to the partition $\{x, y\}, \{z\}$ of $V(T_p)$. Clearly, N_1 has at least two vertices, but so does N_2, by Axiom (D3) applied to $\widetilde{D}_1, \widetilde{D}_2$. In G^+ (which has the edge xy), contract the edge xy. This gives a graph G^{++} that can be obtained from G by splitting p with respect to the partition N_1, N_2 of its neighbors. This split is non-conforming, since the disks $D_1, \ldots, D_4$ violate condition (S1) in the definition of a conforming split. Thus G^{++} is a 3-enlargement of G, as desired.

If $p \neq q$, then from (i) and (ii) above, T_p is either $\{x\}$ or $\{x, x_1\}$. By symmetry, T_q is either $\{y\}$ or $\{y, y_1\}$. If T_p and T_q are both singletons, then clearly $G' = G$ and we are done.

Suppose exactly one of the two branch sets, say T_q, is a singleton, and T_p consists of $\{x, x_1\}$, where x_1 shares a disk D with y in G'. If x_1 and y are not adjacent, then G^+ is a 4-enlargement of G, and we are done. Thus we may assume that x_1 and y are adjacent, and hence by Axiom (D3), they are consecutive in D. Let D_1, D_2 be the two disks in G' containing the edge xx_1. They are both distinct from D, since x and y are not confluent in G'. By Axiom (D3) applied to D_1 and D_2, the vertex x_1 has at least two neighbors in G' other x and y. Now in G^+ (which contains the edge xy), delete the edge x_1y. This gives a graph $\widetilde{G}$ obtained from G by splitting p in the same way as in G', except that y is adjacent to x rather than x_1. Further, it is a non-conforming split, as the disks D, D_1 and D_2 violate condition (S1) in the definition of a conforming split. Thus $\widetilde{G}$, which is isomorphic to a minor of H, is a 3-enlargement of G, and we are done.

Finally, suppose $T_p = \{x, x_1\}$ and $T_q = \{y, y_1\}$, where x shares a disk D'_1 with y_1 and y shares a disk D'_1 with x_1. Let D_1, D_2 be the corresponding disks in G. Since x and y are not confluent in G', D'_1 does not contain y and D'_2 does not contain x. (In particular, D'_1 and D'_2 are distinct.) Apply Lemma 2.4.5 to $\widehat{G} = G'/xx_1$, with the vertices p, y, y_1 in that graph corresponding to p, q, r in the lemma. Thus the (conforming) split of the vertex q in G that produces $\widehat{G}$ is along D_2, and D_2 is one of the disks containing the edge pq in G.

Also, since x and y are not confluent in G', the (conforming) split of p in $\widehat{G}$ that produces G' must be along D_1, and D_1 is the other disk in G containing pq. It now follows that G^+ is a 5-enlargement of G. This finishes the proof of the lemma. $\square$

Lemma 2.7.2 *Let G and H be weakly 4-connected graphs, and let $\mathcal{D}$ be a weak disk system in G with no locally planar extension into H. Let G' be a conforming expansion of G such that H has a minor isomorphic to a 2-enlargement of G'. If $G' \neq G$, then there exists a conforming expansion G'' of G obtained from G' by contracting at least one new edge such that H has a minor isomorphic to an i-enlargement or a weak 9-enlargement of G'' for some $i \in \{2, 6, 7\}$.*

Proof: We may assume that G' is as stated in the lemma, and subject to that, it is minor-minimal. By hypothesis, there are vertices u, v, x, y appearing on a disk C' in G', in that cyclic order, such that H has a minor isomorphic to a graph obtained from G' by adding the edges ux and vy. Let C be the cycle in G corresponding to C'. The minimality of G' implies that every new edge of G' has both ends in $\{u, v, x, y\}$, and hence it belongs to C' by (D3). We may therefore assume that uv is a new edge of G'. We claim that if v is adjacent to x, then the lemma holds. To prove this claim suppose that v and x are adjacent in G', and let $G_1 = G^+ \backslash vx$. If v has degree three in G', then G_1 is isomorphic to a weak 9-enlargement of G'/uv (the new edge is yv; notice that y is not confluent with the edge of G'/uv that is being subdivided by (D3)), and hence the lemma holds. Thus we may assume that v has degree at least four in G'. In that case G_1 is isomorphic to a 4-enlargement of G'/uv, for a graph isomorphic to G_1 can be obtained by a conforming split of the new vertex of G'/uv, not along C', and joining one of the new vertices to y. This proves our claim, and hence we may assume that v is not adjacent to x. By symmetry we may also assume that u is not adjacent to y.

If uv is the only new edge of G', then G' is a 6-enlargement of G, and the lemma holds. Thus we may assume that G' has another new edge, and so that edge must be xy and there are no other new edges. It follows that G' is a 7-enlargement of G, and so the lemma holds. $\square$

Lemma 2.7.3 *Let G and H be graphs, let $\mathcal{D}$ be a weak disk system in G, and let G' be a conforming expansion of G such that H has a minor isomorphic to a 9-enlargement G^+ of G'. If $G' \neq G$, then there exists a conforming expansion G'' obtained from G' by contracting at least one new edge such that H has a minor isomorphic to a 3-enlargement or a weak 9-enlargement of G''.*

Proof: Let $u, x, y \in V(G')$ be such that G^+ is obtained from G' by subdividing the edge xy and joining the new vertex to u, and let f be a new edge of G'. Then $f \neq xy$, for otherwise Lemma 2.4.5 implies that u is confluent with the edge xy, a contradiction. We may assume that f is incident with u, and that contracting f makes the new vertex confluent with the edge xy, for otherwise G^+/f is a weak 9-enlargement of G'/f, and the lemma holds. Hence the other end v of f must share a disk D_1 with the edge xy. Since u is not confluent with xy, D_1 does not contain u. Let D_2 and D_3 be disks shared by u and x, and by u and y, respectively. These three disks are pairwise distinct, since u is not confluent with the edge xy in G'. Now apply Lemma 2.4.5 with x as the vertex p, and u, v as the vertices q, r respectively. It follows that v and x are adjacent, and that D_1 and D_2 are the two disks containing the edge vx. Apply Lemma 2.4.5 again, this time with y in place of x. It follows that the edges vu, vx and vy are covered twice each by the three disks D_1, D_2 and D_3. In particular, D_1 is a triangle.

If $f' \neq f$ is a new edge of G', then by what we have shown about f it follows that f' is incident with u and its other end belongs to a disk D_1' that contains the edge xy. Since D_1 is a triangle consisting of x, y and an end of f, we see that $D_1' \neq D_1$. But the disks that correspond to D_1 and D_1' in $G'/f/f'$ have three vertices in common, contrary to (D3). Thus f is the only new edge of G', and hence $G = G'/f$. Let p be the new vertex of $G = G'/f$.

Since G^+ is a 9-enlargement of G', the graph $G' \backslash \{u, x, y\}$ is connected, and hence v has a neighbor outside $\{u, x, y\}$. (In fact, it must then have at least three neighbors outside $\{u, x, y\}$.) Let z be the new vertex of G^+ created by subdividing the edge xy. The graph $G^+ \backslash vx/xz$ is isomorphic to a graph obtained from G by splitting p into two vertices. This split is non-conforming, since the two disks in G that contain py violate condition (S2) in the definition of a conforming split. Thus H has a minor isomorphic to a 3-enlargement of

G. This finishes the proof of the lemma. $\square$

Lemma 2.7.4 *Let G and H be graphs, let $\mathcal{D}$ be a weak disk system in G, and let G' be a conforming expansion of G such that H has a minor isomorphic to a 8-enlargement G^+ of G'. If $G' \neq G$, then there exists a conforming expansion G'' obtained from G' by contracting at least one new edge such that H has a minor isomorphic to a 3-enlargement, 10-enlargement, or a weak 9-enlargement of G''.*

Proof: Let G^+ be obtained from G' by adding a vertex adjacent to x_1, x_2, x_3, and let f be a new edge of G'. We may assume that upon contracting f the vertices that correspond to x_1, x_2, x_3 belong to a common disk, for otherwise G^+/f is a weak 8-enlargement of G'/f, and the lemma holds. Thus f is incident with at least one of x_1, x_2, x_3, say x_1, and there exists a disk $\mathcal{D}$ in G' that includes y, x_2, x_3, where y is the other end of f.

Apply Lemma 2.4.5 twice, once with x_2 as the vertex p, and next with x_3 as the vertex p. In both applications, let x_1 and y be the vertices q and r respectively. It follows that y is adjacent to x_2 and x_3, and that $yx_1 \in E(D_2 \cap D_3)$, $yx_2 \in E(D \cap D_3)$ and $yx_3 \in E(D \cap D_2)$. Since G^+ is a 8-enlargement of G' the graph $G' \backslash \{x_1, x_2, x_3\}$ is connected, and hence y has degree at least four. Let N be the neighbors of y in G' other than x_1, x_2, x_3. Let G' be obtained from G by splitting x_1 in such a way that the neighborhood of one of the new vertices is N. Then G' is isomorphic to a minor of G^+, and it is a 3-enlargement of G'. Thus the lemma follows from Lemma 2.4.4. $\square$

We are finally ready to state and prove the generalization of Theorem 2.2.1 mentioned earlier, and then deduce Theorem 2.2.1.

Theorem 2.7.5 *Let G and H be weakly 4-connected graphs such that H has a minor isomorphic to G. Let G have a weak disk system $\mathcal{D}$ that has no locally planar extension into H. Then H has a minor isomorphic to an i-enlargement of G, for some $i \in \{1, 2, \ldots, 12\}$.*

Proof: There exists an expansion of G whose subdivision is isomorphic to a subgraph of H. If this expansion is not conforming, then the theorem holds by Lemma 2.4.4, and so we may assume that the expansion is conforming. By Lemma 2.6.5 there exists a conforming expansion G' of G such that H has a minor isomorphic to an i-enlargement G^+ of G' for some $i \in \{1, 2, \ldots, 12\}$. We may choose G' and G^+ such that $|E(G')|$ is minimum. If $i \in \{1, 4, 5, 6, 7\}$, then G^+ is isomorphic to a 1-enlargement of a conforming expansion of G', and the theorem holds by Lemma 2.7.1. If $i \in \{3, 10\}$, then the theorem holds by Lemma 2.4.4. If $i = 12$, then the minimality of G' implies that $G = G'$, and if $i \in \{2, 8, 9.11\}$, then the same conclusion follows from Lemmas 2.7.2, 2.7.4, 2.7.3 and 2.4.1, respectively, using Lemmas 2.6.3 and 2.6.4. Thus the theorem holds. $\square$

Proof of Theorem 2.2.1. Let $G, \mathcal{D}$ and H be as in Theorem 2.2.1. By Theorem 2.7.5 it suffices to show that G has no i-enlargement for $i \in \{10, 11, 12\}$. This is clear for $i = 12$, and follows easily from (D2) when $i = 10$ or $i = 11$, as desired. $\square$

2.8 An Application

In this section, we illustrate an application of Theorem 2.2.1. The projective plane is the non-orientable surface obtained from a closed disk by identifying all pairs of antipodal points on its boundary. Archdeacon [2, 3] proved that a graph H does not embed in the projective plane if and only if it has a minor isomorphic to some graph in an explicitly constructed list of 35 graphs. One might hope that if we assume that H is sufficiently connected, then the list may be shortened. Mohar and Thomas (work in progress) developed a strategy for a proof, but it will be a lengthy project with several intermediate steps. Here we complete one such step: under the assumptions that H is weakly 4-connected and has a minor isomorphic to the Petersen graph, Theorem 2.8.1 below gives a list of eight forbidden minors, each of which are weakly 4-connected.

Figure 2.1 shows these eight graphs (with a vertex-labeling for each of them). All of these graphs, with the exception of F_1' and D_3', appear in the list of 35 forbidden minors for the projective plane. F_1' and D_3', however, are obtained from two graphs in that list (F_1 and D_3, respectively) by splitting exactly one vertex. (The reason we list F_1', D_3' instead of F_1, D_3 is that the latter two graphs are not weakly 4-connected.)

Theorem 2.8.1 *Let H be a weakly 4-connected graph that has a minor isomorphic to the*

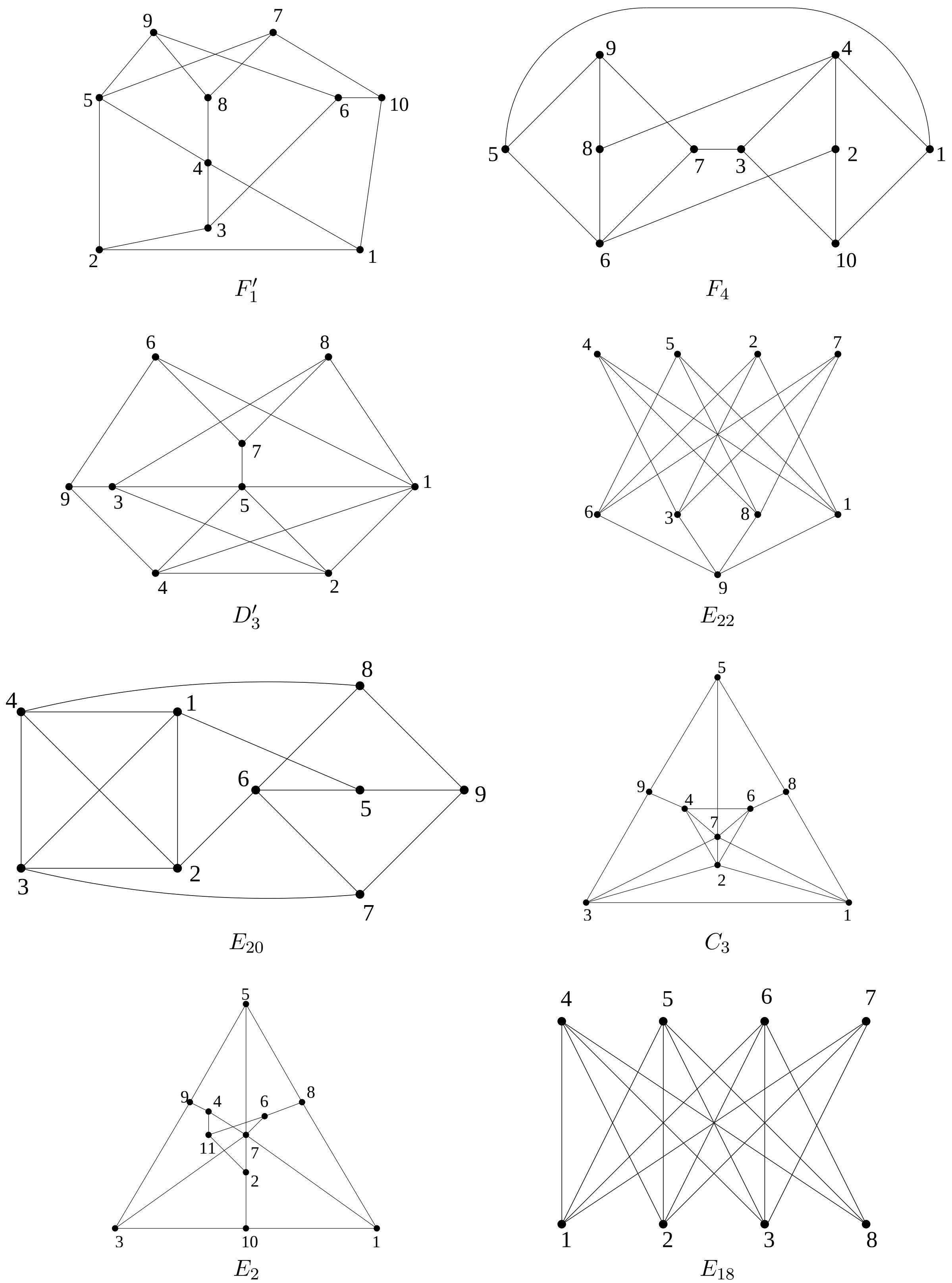

Figure 2.1: The eight graphs of Theorem 2.8.1

Petersen graph. Then H does not embed in the projective plane if and only if it has a minor isomorphic to one of the eight graphs F_1', F_4, D_3', E_{22}, E_{20}, C_3, E_2, or E_{18} shown in Figure 2.1.

Before we derive Theorem 2.8.1 from Theorem 2.2.1, we describe some notation that will be convenient in the proof.

Let P_{10} denote a labeling of the Petersen graph as shown in Figure 2.2. In fact, Figure 2.2 shows an embedding of P_{10} in the projective plane. The disk system $\mathcal{D}$ associated with this embedding consists of the 5-cycles 6-9-7-10-8, 1-5-10-7-2, 4-3-8-10-5, 2-1-6-8-3, 5-4-9-6-1, and 3-2-7-9-4.

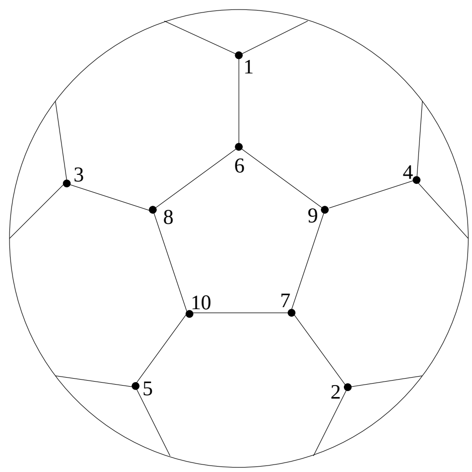

Figure 2.2: One of the two projective-planar embeddings of the Petersen graph

P_{10} has exactly one other embedding in the projective plane. This embedding is distinct from the above embedding, but is isomorphic to it. (An isomorphism of embeddings is an isomorphism τ of the underlying graphs such that a cycle C is facial in one embedding if and only if $\tau(C)$ is facial in the other.) The disk system $\mathcal{D}'$ associated with the second embedding consists of the 5-cycles 1-2-3-4-5, 6-9-4-3-8, 7-10-5-4-9, 8-6-1-5-10, 9-7-2-1-6, and 10-8-3-2-7.

We now describe notation that will let us denote specific enlargements of a (labeled) graph as given by Theorem 2.2.1. Recall the operations 1–9 and the definition of a split, as described in Sections 2.1 and 2.2.

Let G be a graph whose vertices are labeled $1, \ldots, n$. For vertices u, v, the graph

$G+(u,v)$ denotes the graph obtained from G by adding an edge joining u and v (if none existed before). Also, the graph $G*v(N_1)$ denotes the graph obtained by splitting the vertex v, where N_1 is as in the definition of a split. We follow the convention that the vertex v_1 retains the same label as v, while v_2 is assigned the label $n+1$.

Since operations 1–7 are defined in terms of vertex splits and edge additions, the above notation lets us specify i-enlargements for $i = 1, \ldots, 7$. An 8-enlargement of G is specified as $G+(x_1, x_2, x_3)$, where the vertices x_i are as in the definition of operation 8. The new vertex x gets the label $n+1$.

Finally, a 9-enlargement of G is specified as $G+(u, x-y)$, where u, x, y are as in the definition of operation 9. The new vertex obtained by subdividing the edge xy gets the label $n+1$.

2.8.1 Proof of Theorem 2.8.1.

For the backward implication of Theorem 2.8.1, recall that each of the eight graphs specified is either isomorphic to one of the 35 forbidden minors of [3] or is obtained from one of them by splitting a vertex. In particular, none of these eight graphs embed in the projective plane, and so H does not embed either.

For the forward implication, H, by hypothesis, does not embed in the projective plane, and has a minor isomorphic to P_{10}. Clearly, the disk system $\mathcal{D}$ of P_{10} has no locally planar extension to H. Applying Theorem 2.2.1 to $P_{10}, \mathcal{D}$ and H, it is easy to check that H has a minor isomorphic to one of three enlargements, up to isomorphism:

1. a 2-enlargement $Q_1 = P_{10} + (7,8) + (9,10)$
2. an 8-enlargement $Q_2 = P_{10} + (2,4,6)$
3. a 9-enlargement $Q_3 = P_{10} + (1, 3-4)$

Q_2 has a minor isomorphic to E_{18}, as witnessed by the branch sets $\{1,5\}$, $\{3,8\}$, $\{7,9\}$, $\{2\}$, $\{4\}$, $\{6\}$, $\{10\}$, and $\{11\}$. (The order of the branch sets follows that of the corresponding vertex labels in E_{18}, as shown in Figure 2.1.)

Thus we may assume that H has a minor isomorphic to Q_1 or Q_3. The disk system $\mathcal{D}'$ of P_{10} extends in a natural way to disk systems $\mathcal{D}_1, \mathcal{D}_3$ in the enlargements Q_1, Q_3. Thus Q_1, Q_3 each embed (*uniquely*) in the projective plane. The embeddings are shown in Figure 2.3.

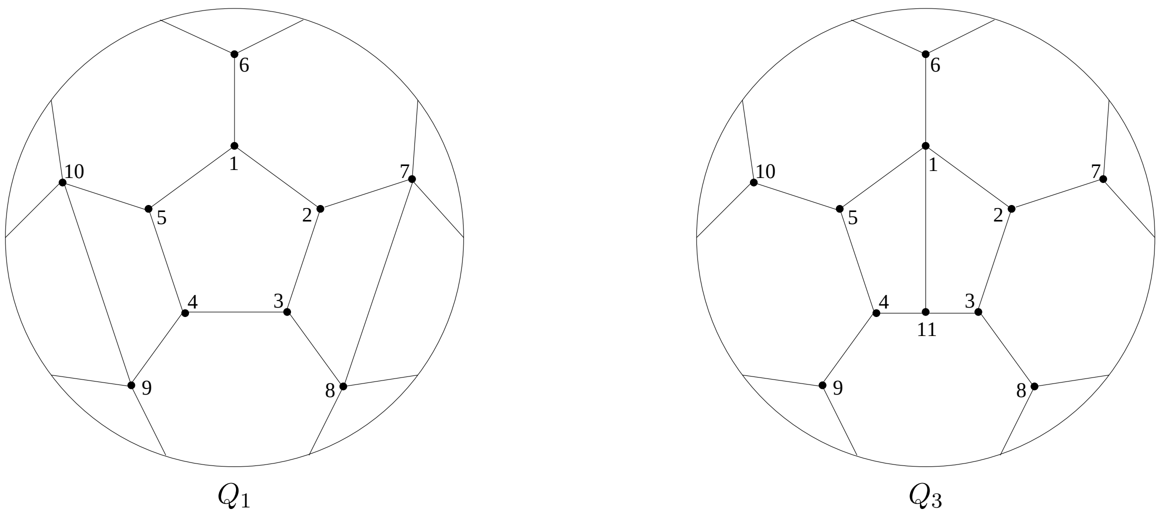

Figure 2.3: The graphs Q_1 and Q_3

We now apply Theorem 2.2.1 to $Q_1, \mathcal{D}_1, H$ and $Q_3, \mathcal{D}_3, H$ and deduce Theorem 2.8.1. This involves a fair amount of case-checking, which is summarized in Tables 2.1 and 2.2. Each row in the tables lists an enlargement of Q_1 or Q_3, along with one of the eight graphs from the list that is a minor of the enlargement. The branch sets in the rightmost column follow the order of the vertex labels of the corresponding graph in the preceding column. For clarity, singleton sets are not enclosed in braces.

Tables 2.1 and 2.2 respectively list all possible enlargements of Q_1 and Q_3 up to isomorphism, with the exception of 8-enlargements and 9-enlargements of Q_1, and 8-enlargements of Q_3. Every 8-enlargement of Q_1 with respect to $\mathcal{D}_1$ has a subgraph isomorphic to Q_2, and thus has a minor isomorphic to E_{18}. Every 8-enlargement of Q_3 with respect to $\mathcal{D}_3$ either has a minor isomorphic to Q_2, or is isomorphic to the 8-enlargement listed in Table 2.2. Finally, every 9-enlargement of Q_1 with respect to $\mathcal{D}_1$ is either isomorphic to the 9-enlargement listed in Table 2.1 or is isomorphic to a 2-enlargement of Q_3 with respect to $\mathcal{D}_3$ (and is thus listed in Table 2.2 instead). This finishes the proof of Theorem 2.8.1. $\square$

Table 2.1: Applying Theorem 2.2.1 to Q_1

Type	Enlargement	Minor	Branch sets of the minor
1	$Q_1{+}(2,10)$	D_3'	$\{2,3\},7,9,8,10,1,5,4,6$
	$Q_1{+}(3,10)$	F_1'	$8,7,2,3,10,1,9,4,5,6$
2	$Q_1{+}(2,8){+}(3,7)$	E_{20}	$2,7,3,8,\{1,6\},9,4,10,5$
	$Q_1{+}(2,4){+}(3,5)$	E_{22}	$2,3,5,4,7,8,10,9,\{1,6\}$
	$Q_1{+}(1,4){+}(3,5)$	F_4	$2,4,5,1,7,9,10,6,8,3$
	$Q_1{+}(1,4){+}(2,5)$	F_4	$1,3,5,2,6,8,10,7,9,4$
	$Q_1{+}(3,9){+}(4,8)$	C_3	$3,4,1,10,7,9,2,5,\{6,8\}$
	$Q_1{+}(3,9){+}(4,6)$	C_3	$3,4,1,10,7,9,2,5,\{6,8\}$
	$Q_1{+}(4,6){+}(8,9)$	E_{20}	$8,7,10,9,3,\{1,2\},5,6,4$
	$Q_1{+}(2,9){+}(6,7)$	D_3'	$\{1,2\},7,10,6,9,3,4,5,8$
	$Q_1{+}(1,9){+}(6,7)$	F_1'	$1,5,4,9,10,3,7,6,8,2$
	$Q_1{+}(1,9){+}(2,6)$	F_4	$1,10,4,9,6,8,3,7,2,5$
	$Q_1{+}(1,7){+}(2,9)$	D_3'	$9,7,8,2,\{1,6\},4,5,10,3$
	$Q_1{+}(1,7){+}(2,6)$	C_3	$1,2,4,8,10,7,5,3,\{6,9\}$
3	$Q_1{*}7(2,10)$	F_1'	$\{1,6\},5,4,9,10,3,7,11,8,2$
	$Q_1{*}8(3,10)$	F_1'	$2,7,11,\{1,6\},9,8,4,5,10,3$
4	$Q_1{*}7(2,9){+}(1,11)$	F_1'	$2,7,11,\{1,6\},9,8,4,5,10,3$
	$Q_1{*}7(2,9){+}(6,11)$	F_1'	$3,4,5,\{8,10\},9,1,7,11,6,2$
	$Q_1{*}7(2,8){+}(3,11)$	F_1'	$8,7,2,3,11,1,9,4,\{5,10\},6$
	$Q_1{*}8(3,7){+}(2,11)$	F_1'	$\{5,10\},1,6,11,2,9,3,8,7,4$
	$Q_1{*}8(3,7){+}(1,8)$	F_1'	$\{5,10\},11,6,1,8,9,3,2,7,4$
	$Q_1{*}8(3,7){+}(5,8)$	F_1'	$\{1,2\},3,4,5,8,9,11,10,7,6$
	$Q_1{*}8(3,6){+}(4,11)$	F_1'	$8,3,\{2,7\},11,4,1,9,10,5,6$
	$Q_1{*}8(3,6){+}(9,11)$	E_{20}	$7,10,9,11,2,\{1,5,6\},4,8,3$
5	$Q_1{*}7(8,10){*}8(3,7){+}(11,12)$	F_1'	$\{1,2\},6,9,11,12,\{3,4\},10,7,8,5$
	$Q_1{*}7(2,8){*}8(7,10){+}(11,12)$	F_1'	$\{3,4\},9,6,12,11,\{1,2\},10,8,7,5$
	$Q_1{*}7(2,9){*}9(4,6){+}(9,11)$	F_1'	$\{3,4\},8,6,9,11,\{1,2\},10,12,7,5$
	$Q_1{*}7(2,8){*}9(4,10){+}(7,9)$	F_1'	$8,\{3,4\},5,10,9,\{1,2\},12,11,7,6$
	$Q_1{*}7(2,9){*}10(9,11){+}(7,12)$	F_1'	$\{1,6\},2,3,\{8,11\},7,4,12,10,9,5$
	$Q_1{*}7(2,8){*}10(5,9){+}(7,10)$	F_1'	$3,\{1,2\},6,8,7,9,10,12,11,\{4,5\}$
6	$Q_1{*}7(2,8){+}(1,11){+}(6,7)$	F_1'	$2,7,6,1,11,8,\{4,9\},5,10,3$
	$Q_1{*}8(3,7){+}(4,11){+}(8,9)$	F_1'	$\{1,6\},5,10,11,4,7,3,8,9,2$
	$Q_1{*}8(3,6){+}(1,11){+}(8,10)$	F_1'	$2,7,11,\{1,6\},9,8,4,5,10,3$
7	$Q_1{*}8(3,7){*}9(4,10){+}(8,12){+}(9,11)$	F_1'	$\{2,7\},10,5,\{1,6\},11,4,8,12,9,3$
9	$Q_1{+}(1,7{-}8)$	F_1'	$2,7,11,\{1,6\},9,8,4,5,10,3$

Table 2.2: Applying Theorem 2.2.1 to Q_3

Type	Enlargement	Minor	Branch sets of the minor
1	$Q_3{+}(2,4)$	F_1'	$1, 11, 3, 2, \{4, 5\}, 8, 9, 7, 10, 6$
	$Q_3{+}(2,5)$	F_1'	$1, 11, 4, 5, \{2, 3\}, 9, 8, 10, 7, 6$
2	$Q_3{+}(1,7){+}(2,6)$	D_3'	$\{3, 8, 11\}, 2, 7, 6, 1, 4, 5, 10, 9$
	$Q_3{+}(1,9){+}(2,6)$	F_1'	$9, 4, 5, 1, \{3, 11\}, 10, 2, 6, 8, 7$
	$Q_3{+}(1,9){+}(6,7)$	F_4	$5, 11, 9, 1, 10, \{3, 8\}, 6, 2, 7, 4$
	$Q_3{+}(2,9){+}(6,7)$	E_{18}	$1, \{3, 8\}, \{4, 9\}, 2, \{5, 10\}, 6, 11, 7$
	$Q_3{+}(1,7){+}(2,9)$	D_3'	$1, 2, \{3, 8\}, 7, \{6, 9\}, 5, 4, 11, 10$
	$Q_3{+}(2,8){+}(3,7)$	F_4	$11, 9, 5, \{1, 6\}, 3, 7, 10, 2, 8, 4$
	$Q_3{+}(2,10){+}(3,7)$	E_{22}	$1, 2, 3, 11, 5, 10, 7, \{4, 9\}, \{6, 8\}$
	$Q_3{+}(2,10){+}(7,8)$	F_4	$3, 7, 10, 8, 11, \{4, 9\}, 5, 6, 1, 2$
	$Q_3{+}(3,10){+}(7,8)$	F_4	$5, 11, 6, 1, 10, 3, 8, 2, 7, \{4, 9\}$
	$Q_3{+}(2,8){+}(3,10)$	F_1'	$2, 8, 6, 1, \{3, 11\}, 9, 10, 5, 4, 7$
	$Q_3{+}(3,9){+}(4,8)$	E_{22}	$\{1, 6\}, 2, 3, 11, 5, \{7, 10\}, 8, 4, 9$
	$Q_3{+}(3,9){+}(8,11)$	F_1'	$11, 4, 5, \{1, 6\}, 9, 10, 3, 2, 7, 8$
	$Q_3{+}(3,4){+}(8,11)$	D_3'	$\{4, 5\}, 11, 8, 1, \{2, 3\}, 9, 7, 10, 6$
	$Q_3{+}(6,11){+}(8,9)$	F_1'	$10, 7, 2, \{3, 8\}, 9, 1, 4, 11, 6, 5$
	$Q_3{+}(3,9){+}(6,11)$	F_1'	$10, 7, 2, \{3, 8\}, 9, 1, 4, 11, 6, 5$
	$Q_3{+}(3,4){+}(6,11)$	D_3'	$\{1, 2\}, 11, 4, 6, \{3, 8\}, 7, 10, 5, 9$
	$Q_3{+}(4,6){+}(8,9)$	F_4	$5, 11, 6, 1, 10, \{3, 8\}, 9, 2, 7, 4$
	$Q_3{+}(3,9){+}(4,6)$	F_4	$5, 11, 6, 1, 10, \{3, 8\}, 9, 2, 7, 4$
	$Q_3{+}(3,6){+}(8,11)$	E_{20}	$2, 1, 3, 11, 7, \{6, 9\}, 8, \{4, 5\}, 10$
	$Q_3{+}(1,3){+}(2,11)$	F_1'	$2, 3, 6, 1, 11, 9, \{8, 10\}, 5, 4, 7$
3	$Q_3{*}1(2,5)$	F_1'	$7, 10, 5, \{1, 2\}, \{3, 8\}, 4, 6, 12, 11, 9$
4	$Q_3{*}1(5,6){+}(10,12)$	F_1'	$12, 1, 5, 10, \{6, 8\}, 4, \{2, 3\}, 7, 9, 11$
	$Q_3{*}1(5,6){+}(8,12)$	F_1'	$9, \{1, 6\}, 5, \{4, 11\}, 12, 10, 2, 3, 8, 7$
	$Q_3{*}1(5,6){+}(1,3)$	F_1'	$10, 8, 6, \{1, 5\}, 3, \{4, 9\}, 2, 12, 11, 7$
6	$Q_3{*}1(5,6){+}(1,7){+}(9,12)$	F_1'	$8, 10, 5, \{1, 6\}, 7, 4, 2, 12, 9, \{3, 11\}$
8	$Q_3 + (2,9,11)$	F_4	$1, 12, 3, 2, \{4, 5\}, 9, \{6, 8\}, 7, 10, 11$
9	$Q_3{+}(8,1{-}11)$	F_1'	$1, 12, 11, \{2, 3\}, \{6, 8\}, 4, 10, 7, 9, 5$
	$Q_3{+}(8,1{-}2)$	F_1'	$9, 4, 5, \{1, 6\}, \{3, 11\}, 10, 2, 12, 8, 7$
	$Q_3{+}(10,1{-}2)$	F_4	$11, 5, 9, \{1, 6\}, \{3, 8\}, 10, 7, 12, 2, 4$
	$Q_3{+}(6,2{-}3)$	F_1'	$2, 12, 6, 1, \{3, 8, 11\}, 9, 10, 5, 4, 7$
	$Q_3{+}(9,2{-}3)$	F_1'	$5, 4, 11, \{1, 6\}, 9, \{3, 8\}, 7, 2, 12, 10$
	$Q_3{+}(3,1{-}6)$	F_4	$2, 11, 12, 1, 7, \{4, 9\}, \{6, 8\}, 5, 10, 3$
	$Q_3{+}(1,3{-}8)$	F_4	$2, 11, 12, 1, 7, \{4, 6, 9\}, 8, 5, 10, 3$
	$Q_3{+}(2,6{-}8)$	F_4	$3, 7, 12, \{8, 10\}, 11, \{4, 9\}, 6, 5, 1, 2$
	$Q_3{+}(7,6{-}8)$	F_4	$11, 9, 5, \{1, 6\}, \{2, 3\}, 7, 10, 12, 8, 4$
	$Q_3{+}(3,7{-}9)$	F_4	$5, 11, 9, \{1, 6\}, 10, \{3, 8\}, 12, 2, 7, 4$
	$Q_3{+}(8,7{-}9)$	F_4	$5, 11, 9, \{1, 6\}, 10, \{3, 8\}, 12, 2, 7, 4$
	$Q_3{+}(1,7{-}10)$	E_2	$2, 9, 12, 11, 5, 8, \{1, 6\}, 3, 7, 4, 10$
	$Q_3{+}(6,7{-}10)$	E_2	$2, 9, 12, 11, 5, 8, \{1, 6\}, 3, 7, 4, 10$

CHAPTER III

SIX-REGULAR GRAPHS IN $\mathcal{F}_6$

3.1 Preliminaries

We begin with some basic notation and observations that will be used in the proof of this chapter's main theorem. Refer to Section 1.2.2 for a discussion of the theorem and its motivation, in the context of Jorgensen's conjecture (Conjecture 1.2.1).

Let v be a vertex in a graph G, and X be a set of vertices in G, that may or may not contain v. We define $N_X(v)$ as the set of vertices in X that are adjacent to v in G, and $\deg_X(v)$ as the size of this set.

A vertex of G with degree less than 6 is called a *low-degree* vertex.

$\mathcal{F}_6$ is defined to be the family of graphs G such that G has at least 7 vertices, and the set of low-degree vertices in G forms a clique.

We are now ready to state the main theorem of this chapter:

Theorem 3.1.1 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Then G has a minor isomorphic to K_6.*

We now make four easy observations that will be used repeatedly in the proof of Theorem 3.1.1. In the following discussion, we shall denote X to be a clique in G of maximum size. For a vertex $x \in X$, let $N'(x)$ be the set of neighbors of x in $V(G) \setminus X$. Let Y be the union of $N'(x)$ over all vertices $x \in X$. For a vertex $y \in Y$, $\deg_X(y) \geq 1$.

For the following observations, let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Also, assume that G has at least eight vertices. (Clearly, if it has seven vertices, then it must be isomorphic to K_7 and thus Theorem 3.1.1 holds trivially for it.) Let X and Y be defined as above.

Observation 1 *Any two adjacent vertices u, v in G must have at least two common neighbors that are mutually non-adjacent. If u or v (or both) are in the clique X, then at least* *one of those common neighbors must be outside X.*

Observation 2 *Let the clique X be of size 4, with $x \in X$ and $y_1 \in Y$ such that x is the unique neighbor of y_1 in X. Let y_2 and y_3 be the other two neighbors of x outside X. Then y_1, y_2, y_3 span precisely the two edges y_1y_2 and y_1y_3.*

Proof: This follows by applying Observation 1 to x, y_1. $\square$

Observation 3 *Let the clique X be of size 4. No vertex $x \in X$ is adjacent to two vertices $y_1, y_2 \in Y$ such that $\deg_X(y_1) = \deg_X(y_2) = 1$.*

Proof: This follows immediately from Observation 2.

Recall that a *separation* of a graph G is a pair (A, B) of subsets of $V(G)$ such that $A \cup B = V(G)$, and there is no edge between $A - B$ and $B - A$. The separation is called *proper* or *non-trivial* if both $A - B$ and $B - A$ are non-empty; otherwise, it is called *trivial*. The *order* of the separation is defined as $|A \cap B|$. A separation of order $\leq k$ is called a *$\leq k$-separation*.

Observation 4 *G does not have a separation (A, B) such that (i) both $A - B$ and $B - A$ are non-empty, and (ii) $G[A]$ has a clique minor rooted at the vertices of $A \cap B$ (that is, a clique minor in which each vertex of $A \cap B$ is contained in a distinct branch set).*

Proof: This follows by noting that adding to $G[B]$ all possible edges spanning $A \cap B$ yields a graph in which the low-degree vertices form a clique. Further, the graph has at least 7 vertices, and is thus in $\mathcal{F}_6$. Thus from the minor-minimality of G it follows that a separation satisfying conditions (i) and (ii) cannot exist in G. $\square$

The definition of $\mathcal{F}_6$ is motivated by a desire to relax 6-connectivity. (Refer to Section 1.2.2 for a relevant discussion.) However, on the face of it, the definition of $\mathcal{F}_6$ does not say anything directly about separations (and hence about connectivity). Observation 4 above realizes the connectivity implications of that definition.

For $Z \subseteq V(G)$, $G[Z]$ denotes the subgraph *induced* by Z, that is, the subgraph consisting of Z and all edges with both ends in Z. A subgraph of G is said to be *induced* if it is induced by its vertex set.

Let S be a subgraph of a graph G. An *S-bridge of G* is a subgraph $\mathcal{B}$ of G such that either $\mathcal{B}$ consists of a unique edge of $E(G) - E(S)$ and its ends, where the ends belong to S, or $\mathcal{B}$ consists of a component J of $G \backslash V(S)$ together with all edges from $V(J)$ to $V(S)$ and all their ends. For an S-bridge $\mathcal{B}$, the vertices of $\mathcal{B} \cap S$ are called the *attachments* of B.

For $Z \subseteq V(G)$, a Z-bridge of G refers to an S-bridge as above, where S is the subgraph $G[Z]$ induced by Z.

When we say that G has a *rooted subgraph isomorphic to* a graph H shown in some figure, we mean that G has a subgraph isomorphic to H, with the convention that solid vertices in the figure are not incident to any more edges in G than the ones shown. We use vertex labels in the figures to also denote the corresponding vertices in G (relative to some fixed copy of H in G).

A *society* (G, Ω) consists of a graph G, and a cyclic order Ω on a subset $V(\Omega)$ of $V(G)$. The society (G, Ω) is said to have a *cross* if G has two vertex-disjoint paths with ends (s_1, t_1) and (s_2, t_2) respectively, such that s_1, s_2, t_1, t_2 appear in Ω, in the given cyclic order. The vertices s_1, s_2, t_1, t_2 are called the *feet* of the cross.

Theorem 3.1.2 ([48, 49, 54]) *Let (G, Ω) be a society such that G has no ≤ 3-separation (A, B) with $V(\Omega) \subseteq A$. Then exactly one of the following outcomes hold:*

1. *G can be drawn in the plane with the vertices in $V(\Omega)$ on the boundary of the infinite face, in the cyclic order given by Ω*

2. *(G, Ω) has a cross*

3.1.1 Outline of the Main Proof

If G is not six-regular, then deleting an edge incident to a vertex of degree ≥ 7 creates at most one low-degree vertex, and hence the resulting graph is also in $\mathcal{F}_6$, contradicting the minimality of G. Hence it follows that G is *6-regular*. Also, we may assume that the size of a maximum clique is ≥ 3 by Observation 1, and ≤ 5 (otherwise Theorem 3.1.1 holds trivially). The proof of the theorem is split into cases depending on the clique number (that is, size of the largest clique) of G. The main lemmas 3.1.3, 3.2.1 and 3.3.2 handle these cases, with subcases being handled by other lemmas of their own. The subsidiary lemmas usually follow the main lemma that they serve.

Lemma 3.1.3 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let G have a maximum clique X of size 5. Then G has a minor isomorphic to K_6.*

Proof: Let $x_1, \ldots, x_5$ be the vertices of X, and let y_1, y_2 be the two neighbors of x_1 outside X. Applying Observation 1 to each pair of the form x_1, x_i for $i = 2, \ldots, 5$, it follows that each vertex in X is adjacent to y_1 or y_2 (or both). Further, applying Observation 1 to x_1 and y_1, it follows that y_1 and y_2 are adjacent. But then contracting y_1y_2 in G gives a K_6 minor. □

3.2 When G has Clique Number 4

Lemma 3.2.1 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let G have a maximum clique X of size 4. Then G has a minor isomorphic to K_6.*

Proof: Let $x_1, \ldots, x_4$ be the vertices of X. Since G has no clique of size 5, every vertex in Y is incident with at most 3 vertices in X. Each vertex in X has exactly three neighbors outside X, and hence there are 12 X-Y edges. Thus there are at most four vertices y such that $\deg_X(y) = 3$.

If there are exactly four such vertices, then we claim that G must have a K_6 minor. Indeed, G cannot have a vertex outside $X \cup Y$, otherwise Observation 4 applied to the separation $(X \cup Y, V(G) - X)$ yields a contradiction. Thus $V(G) = X \cup Y$. Since G is six-regular, it follows that it is isomorphic to the graph obtained from K_8 by removing a perfect matching. Thus it has a K_6 minor.

Thus the number of vertices y with $\deg_X(y) = 3$ is three, two, one or zero. By Lemmas 3.2.3, 3.2.4, 3.2.7 and 3.2.10, it follows that G has a minor isomorphic to K_6. □

The rest of this section is devoted to the subsidiary lemmas referenced in the above proof.

Lemma 3.2.2 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let G have a largest clique X of size 4. No two vertices y_1, y_2 share the same set of three neighbors in X.*

Suppose $y_1, y_2 \in Y$ are both adjacent to x_1, x_2 and x_3. Since G has no clique of size 5, y_1 and y_2 are non-adjacent. For $i = 1, 2, 3$, let y'_i be the unique neighbor of x_i in Y, other than y_1 and y_2. (Note that the vertices y'_i may not all be distinct.) Applying Observation 1 in turn to y_1, x_i for $i = 1, 2, 3$, we conclude that y_1 must be adjacent to each y'_i. By the same argument, y_2 must be adjacent to each y'_i, for $i = 1, 2, 3$. Further, applying Observation 3 to x_4, it follows that x_4 must be adjacent to at least two *distinct* vertices y'_i and y'_j. But then contracting the edges $y_1y'_i$ and $y_2y'_j$ gives a K_6 minor in G. □

Lemma 3.2.3 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Let G have a largest clique X of size 4, such that there are exactly three vertices $y \in Y$ with $\deg_X(y) = 3$. Then G has a minor isomorphic to K_6.*

Let y_1, y_2, y_3 be the three vertices with the given property. Without loss of generality, we may assume: y_1 is adjacent to x_4, x_1, x_2; y_2 is adjacent to x_4, x_2, x_3; y_3 is adjacent to x_4, x_1, x_3. For $i = 1, 2, 3$, let y'_i be the unique neighbor of x_i in Y, other than y_1, y_2, y_3. (Note that the vertices y'_i may not all be distinct.) In fact, we claim that they cannot all be distinct. Applying Observation 2 to y'_1 and y'_3, we conclude that y_3 can be adjacent neither to y_1 nor y_2. But then applying Observation 1 to x_4y_3 yields a contradiction. Thus the vertices y'_i are not all distinct. (They cannot all be the same either, because of the hypothesis of the lemma.)

Now suppose that $y'_2 = y'_3 = y'$, and that y'_1 is distinct from that vertex. Applying Observation 2 to x_1, y'_1, we conclude that y'_1 is adjacent to both y_1 and y_3, and that y_1, y_3 are non-adjacent. Applying Observation 1 in turn to x_4y_3 and x_2y', it follows respectively that y_3 is adjacent to y_2 and y' is adjacent to y_1. Thus G has a rooted subgraph isomorphic to the graph H_1 shown in Figure 3.1. But then contracting the edges x_2y_2, x_3y' and $y_1y'_1$ yields a K_6 minor in G.

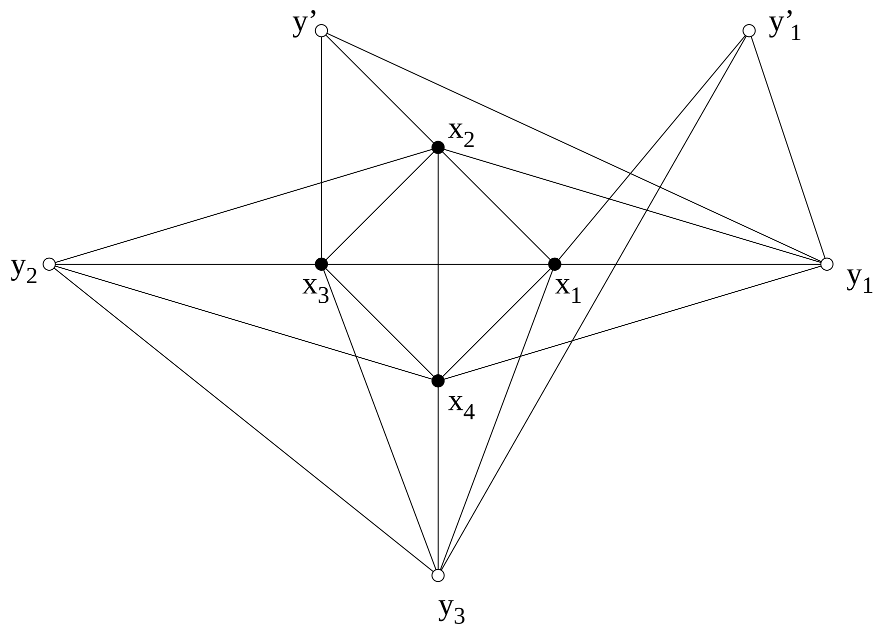

Figure 3.1: Graph H_1

Finally, suppose that $y_1' = y_3' = y'$, and that y_2' is distinct from that vertex. Applying Observation 1 in turn to $x_1 y'$ and $x_3 y'$, it follows that y' is adjacent to y_1 and y_2 respectively. Again, applying Observation 1 to $x_4 y_3$, it follows that y_3 must be adjacent to at least one vertex among y_1 and y_2, say y_1. But then contracting the edges $y_1 y_3$ and $y_2 y'$ yields a K_6 minor in G. $\square$

Lemma 3.2.4 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Let G have a largest clique X of size 4, such that there are exactly two vertices $y \in Y$ with $\deg_X(y) = 3$. Then G has a minor isomorphic to K_6.*

Proof: By Lemma 3.2.2, without loss of generality, we may assume that there exist $y_1, y_2 \in Y$, with y_1 adjacent to x_1, x_2, x_3 and y_2 adjacent to x_1, x_4, x_3. Applying Observation 1 to x_2, x_4, we conclude that there exists y_3 adjacent to x_2, x_4 (and thus distinct from y_1, y_2).

We claim that there cannot exist another vertex y_4 that is adjacent to both x_2 and x_4. Suppose the contrary. Applying Observation 1 in turn to $y_3 x_2$ and $y_3 x_4$, it follows that y_3 is respectively adjacent to y_1 and y_2. By a similar argument, y_4 is also adjacent to y_1 and y_2. But then G has a minor isomorphic to K_6. This proves the claim.

Thus there exist vertices y_4 and y_5, distinct from each other and from y_1, y_2, y_3, and respectively adjacent to x_2 and x_4. We now split the argument into cases:

Case 1: $\deg_X(y_4) = \deg_X(y_5) = 1$.

Applying Observation 2 to x_2y_4, it follows that y_4 is adjacent to y_1, y_3 (and that y_1 and y_3 are non-adjacent). Similarly, y_5 is adjacent to y_2 and y_3.

Now, suppose that x_1 and x_3 have a common neighbor y_6 distinct from y_1, y_2, so that Y consists of the six distinct vertices $y_1, \ldots, y_6$. Thus G has a rooted subgraph isomorphic to the graph H_2 shown in Figure 3.2. By Lemma 3.2.5, we are done.

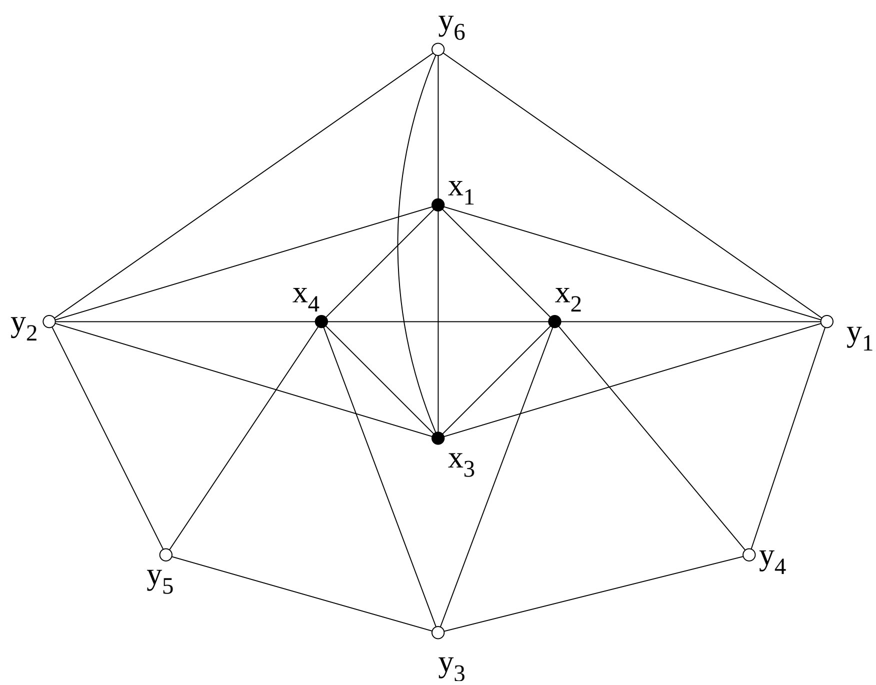

Figure 3.2: Graph H_2

Thus x_1 and x_3 have neighbors y_6 and y_7 respectively, that are distinct from each other and from $y_1, \ldots, y_5$. Applying Observation 2 in turn to x_1y_6 and x_3y_7, it follows that G has a rooted subgraph isomorphic to the graph H_3 shown in Figure 3.3. By Lemma 3.2.6, we are done. This finishes Case 1.

Case 2: Among y_4 and y_5, one vertex has exactly two neighbors in X and the other vertex has exactly one neighbor in X. Without loss of generality, we may assume that y_4 is adjacent to x_2 and x_3, whereas y_5, as before, is adjacent to x_4 (and has no other neighbor in X). This means that there is a vertex y_6 whose unique neighbor in X is x_1,

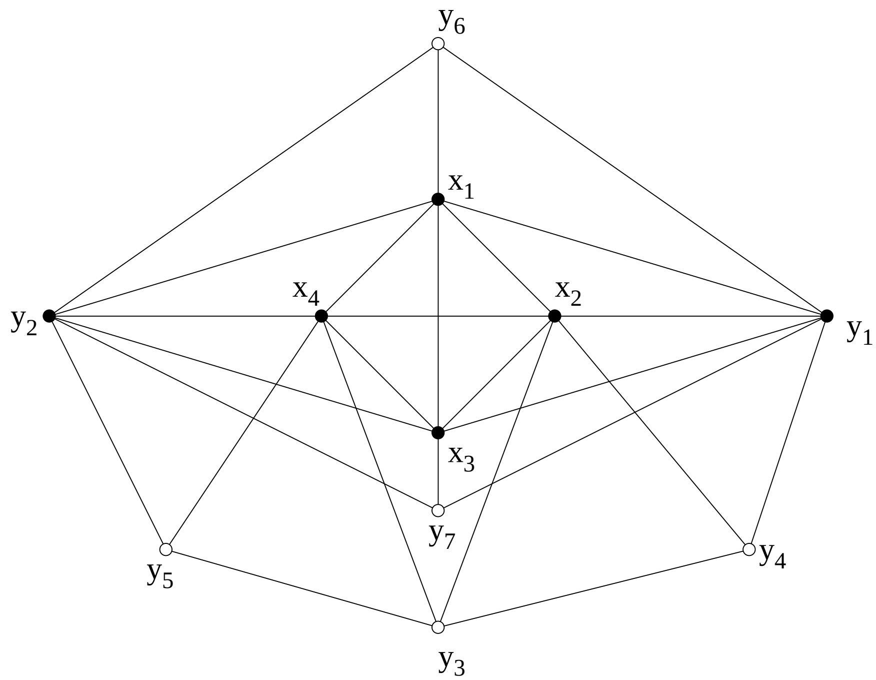

Figure 3.3: Graph H_3

and $Y = \{y_1, \ldots, y_6\}$. Applying Observation 2 to x_4y_5, it follows that y_5 is adjacent to y_2 and y_3. Applying Observation 2 to x_1y_6, it follows that y_6 is adjacent to y_1 and y_2. Finally, applying Observation 1 to x_3y_4, we conclude that y_4 must be adjacent to y_2. Thus G has a rooted subgraph isomorphic to the graph H_4 shown in Figure 3.4. We may assume that the edge y_1y_3 does not exist (otherwise contracting the edges y_1y_3 and y_2y_4 yields a K_6 minor). Since y_3 has degree 6 in G, there must be a vertex in $V(G) - X - Y$. Now the vertices y_1, y_3, y_4, y_5, y_6 separate $X \cup \{y_2\}$ from the rest of G, and the resulting separation violates Observation 4. This finishes Case 2.

Case 3: Both y_4 and y_5 have exactly two neighbors in X. Without loss of generality, we may assume that y_4 is adjacent to x_2, x_3 and y_5 is adjacent to x_4, x_1. Applying Observation 1 in turn to x_1y_5 and x_3y_4, it follows that y_5 is adjacent to y_1 and y_4 is adjacent to y_2. Further, applying Observation 1 in turn to x_2y_4 and x_4y_5, it follows that both y_4 and y_5 are adjacent to y_3. But then contracting the edges y_1y_5 and y_2y_4 yields a minor isomorphic to K_6. This finishes Case 3, and the proof of the lemma. □

Lemma 3.2.5 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such*

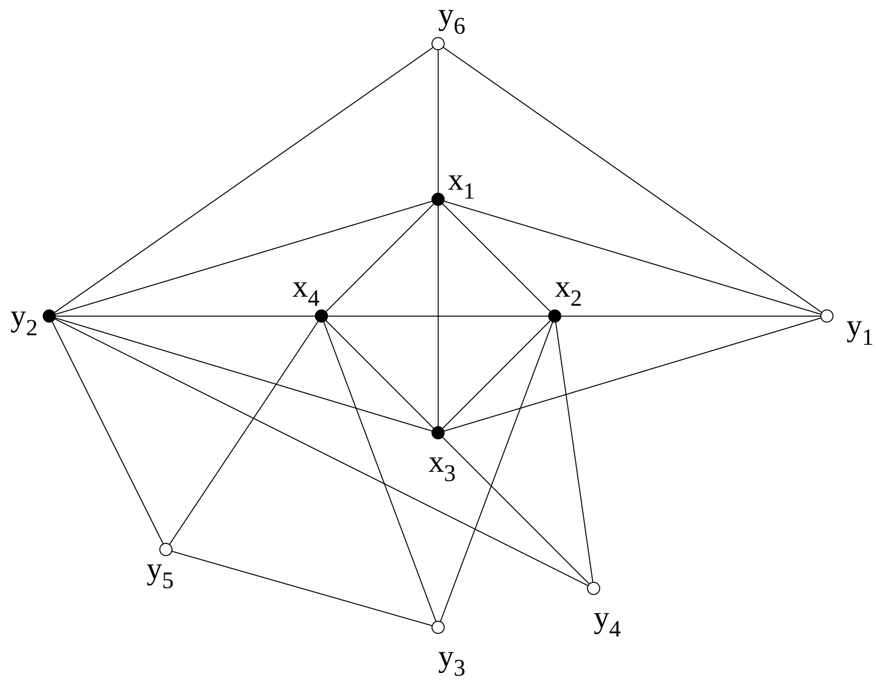

Figure 3.4: Graph H_4

that it has a rooted subgraph isomorphic to the graph H_2 shown in Figure 3.2. Then G has a minor isomorphic to K_6.

Proof: By Lemma 3.1.3, we may assume that G has no clique of size 5. Applying Observation 1 in turn to y_1y_6 and y_1y_4, it follows that one (or both) of the following outcomes happen:

1. y_6 is adjacent to y_4

2. y_1, y_4, y_6 have a common neighbor z_1 outside $X \cup Y$

By symmetry, we also get one (or both) of the following outcomes:

A. y_6 is adjacent to y_5

B. y_2, y_5, y_6 have a common neighbor z_2 outside $X \cup Y$

First, suppose either outcome 1 or outcome A (or both) happen. Without loss of generality, we may assume that outcome 1 happens. If y_1 is adjacent to y_3 or y_5, then G has a minor isomorphic to K_6. Also, y_1 is not adjacent to y_2 (by applying Observation 1 to x_1y_6, for instance). Thus y_1 has a neighbor outside $X \cup Y$. Let $\mathcal{B}$ be the $(X \cup Y)$-bridge of G that contains that vertex. Suppose $\mathcal{B}$ does not attach at y_3 or y_5. Since y_6 is either adjacent to y_5 or has a common neighbor z_2 with it, $\mathcal{B}$ cannot attach at y_6 either. But then the vertices y_1, y_2, y_4 separate $\mathcal{B}$ from the rest of G, and the resulting separation violates Observation 4. Thus $\mathcal{B}$ must attach at y_3 or y_5 (or both). But then G has a minor isomorphic to K_6 (with branch sets given by $\mathcal{B} \cup \{y_1, y_3, y_5\}, \{y_2, y_6, y_4\}, \{x_1\}, \ldots, \{x_4\}$).

Thus we may assume that both outcome 2 and outcome B happen, and that y_6 is adjacent to neither y_4 nor y_5. Suppose that $z_1 = z_2 = z$. If y_6 is adjacent to y_3, then G has a K_6 minor. Since y_6 has degree 6 in G, it has a neighbor outside $X \cup Y \cup \{z\}$, and let $\mathcal{B}$ be the $(X \cup Y)$-bridge of G that contains that neighbor. If $\mathcal{B}$ attaches at y_3, then G has a K_6 minor as before. Thus the vertices y_4, y_5, y_6, z separate $\mathcal{B}$ from the rest of G, and the resulting separation violates Observation 4.

Hence z_1, z_2 must be distinct. Applying Observation 1 to $y_6 z_1$, it follows that z_1 is adjacent to z_2. Let $P = X \cup \{y_1, y_2, y_6\}$. The five vertices z_1, z_2, y_5, y_3, y_4 separate P from the rest of G, and the graph induced by those five vertices is a 5-cycle, in that order. We claim that $G \setminus P$ is not a planar graph. Indeed, if the number of vertices in $G \setminus P$ is n, the number of edges is $\frac{1}{2}(6(n-5)+20) = 3n-5$. Let $G_0 = G \setminus P$ and $\Omega = z_1, z_2, y_5, y_3, y_4$ (in that cyclic order).

We claim that G_0 has no ≤ 3-separation (A, B) with $V(\Omega) \subseteq A$. Suppose it does; choose (A, B) as above such that its order k is minimal. Then by Menger's theorem, there exist k vertex disjoint paths from $V(\Omega)$ to $A \cap B$. It now follows easily that the separation $(A \cup P, B)$ in G violates Observation 4. This proves our claim.

Thus by Theorem 3.1.2, the society (G_0, Ω) has a cross. By symmetry, there are three possibilities to consider for the cross, and it is easy to check that G has a K_6 minor in each of those cases. $\square$

Lemma 3.2.6 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it has a rooted subgraph isomorphic to the graph H_3 shown in Figure 3.3. Then G has a minor isomorphic to K_6.*

Proof: By Lemma 3.1.3, we may assume that G has no clique of size 5. Applying Observation 1 to y_1y_4, it follows that y_4 is adjacent to either y_6 or y_7 (or both). By symmetry, y_5 is also adjacent to either y_6 or y_7 (or both). If there are two disjoint edges between $\{y_4, y_5\}$ and $\{y_6, y_7\}$, then G has a K_6 minor. If not, we may assume, without loss of generality, that y_6 is adjacent to both y_4 and y_5, and y_7 is adjacent to neither. Further, applying Observation 1 to y_1y_7, it follows that y_7 is adjacent to y_6. Since y_4 has degree 6, it must have a neighbor outside $X \cup Y$. But then $\{y_3, y_4, y_5, y_7\}$ separates $X \cup \{y_1, y_2, y_6\}$ from the rest of G, and the resulting separation violates Observation 4. □

Lemma 3.2.7 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Let G have a largest clique X of size 4, such that there is exactly one vertex $y_{1\,2\,3} \in Y$ with $\deg_X(y_{1\,2\,3}) = 3$. Then G has a minor isomorphic to K_6.*

Proof: Let $y_{1\,2\,3}$ be adjacent to x_1, x_2, x_3. By the hypothesis of the lemma, and by Observation 1 applied in turn to x_ix_4 for $i = 1, 2, 3$, it follows that there are vertices $y_{i\,4}$ adjacent to precisely x_i, x_4 (and no other vertex in X). Now for $i = 1, 2, 3$, x_i has a unique neighbor y_i distinct from $y_{1\,2\,3}, y_{1\,4}, y_{2\,4}, y_{3\,4}$. (Note that the vertices y_i may not all be distinct.)

First, suppose that the vertices y_i are all distinct. Applying Observation 2 in turn to x_iy_i, for $i = 1, 2, 3$, it follows that y_i is adjacent to both $y_{1\,2\,3}$ and $y_{i\,4}$ (and that $y_{1\,2\,3}, y_{i\,4}$ are non-adjacent). Thus G has a rooted subgraph isomorphic to the graph H_5 shown if Figure 3.5. By Lemma 3.2.8, we are done.

Hence we may assume, without loss of generality, that $y_1 = y_2 = y_{1\,2}$, and y_3 is distinct from that vertex. Applying Observation 1 in turn to $x_1y_{1\,2}$ and $x_2y_{1\,2}$, it follows that $y_{1\,2}$ is adjacent to $y_{1\,4}$ and $y_{2\,4}$. Applying Observation 2 to x_3y_3, it follows that y_3 is adjacent to $y_{1\,2\,3}$ and $y_{3\,4}$ (and that the latter two vertices are non-adjacent).

Suppose $y_{1\,2\,3}$ is not adjacent to $y_{1\,2}$. Then apply Observation 1 in turn to $x_1y_{1\,2\,3}, x_2y_{1\,2\,3}, x_4y_{3\,4}$; it follows respectively that $y_{1\,2\,3}$ is adjacent to both $y_{1\,4}$ and $y_{2\,4}$, and $y_{3\,4}$ is adjacent to $y_{1\,4}$ or $y_{2\,4}$ (or both). But then G has a K_6 minor. Hence we may assume that $y_{1\,2\,3}$ is adjacent to $y_{1\,2}$.

Thus G has a rooted subgraph isomorphic to the graph H_6 shown in Figure 3.6. By

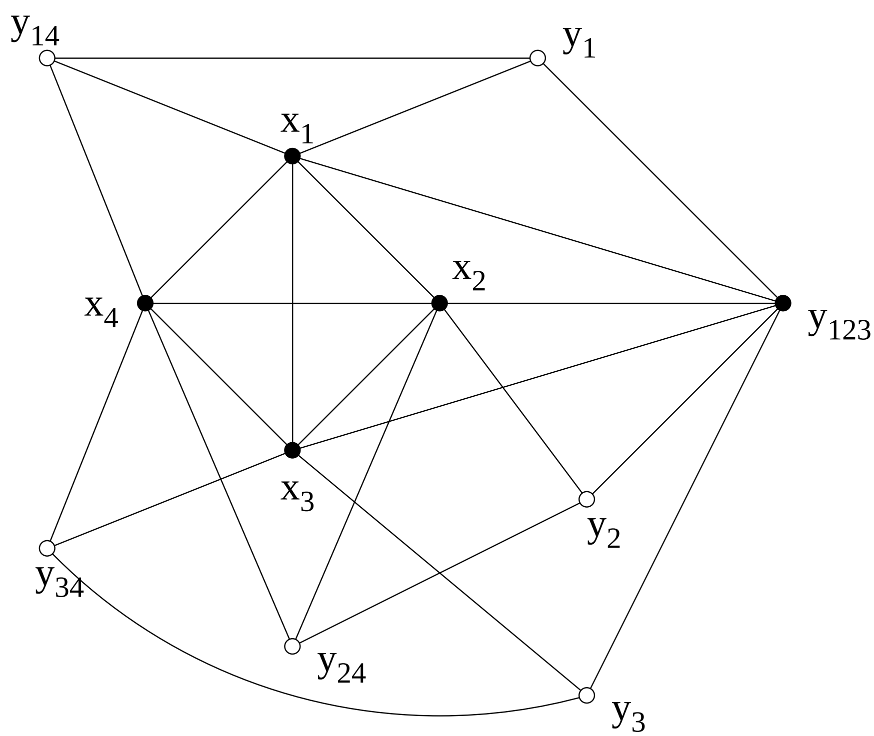

Figure 3.5: Graph H_5

Lemma 3.2.9, we are done. □

Lemma 3.2.8 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it has a rooted subgraph isomorphic to the graph H_5 shown in Figure 3.5. Then G has a minor isomorphic to K_6.*

Proof: By Lemma 3.1.3, we may assume that G has no clique of size 5. Let $P = \{y_i \ : \ i = 1,2,3\}$, $Q = \{y_{i\,4} \ : \ i = 1,2,3\}$, $X = \{x_1, \ldots, x_4\}$ and $Y = \{y_{1\,2\,3}\} \cup P \cup Q$. For $i = 1,2,3$, apply Observation 1 to each of $y_i y_{1\,2\,3}$, and to each of $y_{i\,4} x_4$. It then follows that the graphs induced by P and Q are both connected.

First, suppose that $V(G) = X \cup Y$. Since G is 6-regular, $G[P \cup Q]$ is 4-regular. Hence each vertex in P is adjacent to at least two vertices in Q, and vice-versa. Since $G[P]$ is connected, we may assume, without loss of generality, that y_2 is adjacent to y_3. Now y_1 is adjacent to at least one vertex among $y_{2\,4}$ or $y_{3\,4}$, say $y_{3\,4}$. Further, $y_{1\,4}$ is adjacent to at least one vertex among y_2, y_3. But then G has a K_6 minor (with branch sets given by $\{y_{1\,2\,3}, y_1, y_{3\,4}\}, \{y_{2\,4}, y_2, y_3, y_{1\,4}\}, \{x_1\}, \ldots, \{x_4\}$.

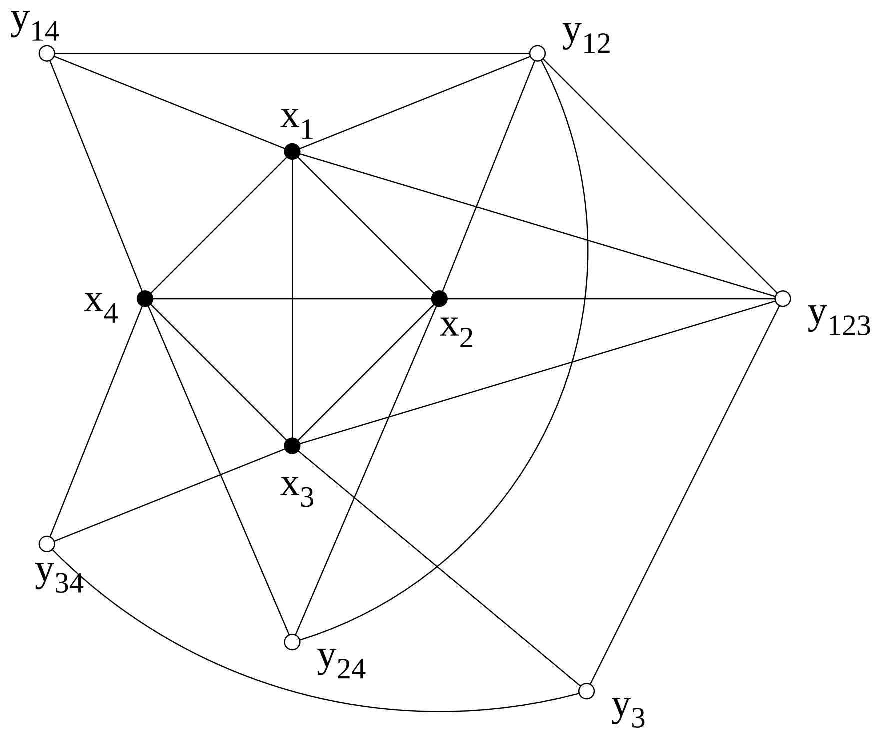

Figure 3.6: Graph H_6

Thus we may assume that G has a vertex outside $X \cup Y$. Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G. We claim that $\mathcal{B}$ has at least 5 attachments in $P \cup Q$. Suppose the contrary, and let B_0 be the set of its attachments. Consider the separation $(G \setminus (V(\mathcal{B}) - B_0), V(\mathcal{B}))$. If there exists a matching between B_0 and X that saturates B_0, then contracting it violates Observation 4. Hence, we may assume, without loss of generality, that $B_0 = \{y_1, y_2, y_{14}, y_{24}\}$. Since $G[P]$ is connected, y_3 is adjacent to at least one vertex among y_1 and y_2, say y_1. But then contracting the sets $\{x_4, y_{24}\}, \{y_1, y_3, y_{34}\}, \{y_2, x_2, x_1\}$ still violates Observation 4. This proves the claim (that $|B_0| \geq 5$).

Now since $G[P]$ is connected, without loss of generality, y_2 is adjacent to both y_1 and y_3. Since $G[Q]$ is also connected, we thus get the following two cases:

Suppose y_{24} is adjacent to both y_{14} and y_{34}. Since B_0 has at least 5 vertices, it contains either $\{y_1, y_{34}\}$ or $\{y_3, y_{14}\}$ (or both). In any case, it is easy to check that G has a K_6 minor.

Alternatively, suppose y_{14} is adjacent to both y_{24} and y_{34}. If $\mathcal{B}$ does not attach at y_1, then it attaches at all other vertices in $P \cup Q$, and G has a K_6 minor. If $\mathcal{B}$ attaches at y_1, then it also attaches at y_{24} or y_{34} (or both). In either case, G has a K_6 minor. This finishes the proof of the lemma. $\square$

Lemma 3.2.9 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it has a rooted subgraph isomorphic to the graph H_6 shown in Figure 3.6. Then G has a minor isomorphic to K_6.*

Proof: By Lemma 3.1.3, we may assume that G has no clique of size 5. Suppose that y_{123} is adjacent to y_{24}. Applying Observation 1 to $y_{123}y_{24}$, it follows that y_{24} is adjacent to y_3. Since $\deg(y_{24}) = 6$, y_{24} is not adjacent to at least one vertex among y_{14} and y_{34}, say the former. Applying Observation 1 to x_4y_{14}, it follows that y_{14} is adjacent to y_{34}. But then G has a K_6 minor. Hence we may assume that y_{123} is not adjacent to y_{24}. By a similar argument, we may also assume that y_{123} is not adjacent to y_{14}.

From Observation 2 applied to x_3y_3, y_{123} is not adjacent to y_{34}. Now this means that y_{123} has a neighbor outside $X \cup Y$. Applying Observation 1 to $y_{123}y_{12}$, we get one of the following two cases:

Case 1: The vertex y_{12} is adjacent to y_3. But then consider the non-trivial separation $(X \cup Y, V(G) \setminus (X \cup \{y_{12}\}))$. Contracting the edges $y_{12}x_1, y_{34}x_4, y_{24}x_2, y_{12}y_3, y_{123}x_3$, it follows that the separation violates Observation 4, a contradiction.

Case 2: The vertices y_{123} and y_{12} have a common neighbor z_1 outside $X \cup Y$. Applying Observation 1 to $y_{123}y_3$, it follows that y_3 is adjacent to z_1. Applying Observation 1 to $y_{12}y_{14}$, it follows that y_{14} is adjacent to either z_1 or y_{24} (or both). By symmetry, y_{24} is adjacent to either z_1 or y_{14} (or both). Thus, if y_{14}, y_{24} are non-adjacent, then z_1 must be adjacent to both of them. Further, applying Observation 1 to x_4y_{34}, it follows that y_{34} is adjacent to at least one vertex among y_{14}, y_{24}, and then G has a K_6 minor. Hence we may assume that y_{14}, y_{24} are adjacent.

Now if z_1 has no neighbor outside $X \cup Y$, then it must be adjacent to y_{14} and y_{24}, and again, G has a K_6 minor. Thus z_1 must have a neighbor outside $X \cup Y$. Let $\mathcal{B}$ be an $(X \cup Y \cup \{z_1\})$-bridge of G containing such a neighbor.

If $\mathcal{B}$ attaches at both y_{14} and y_{24}, then G has a K_6 minor as before. Hence, without loss of generality, we may assume that it does not attach at y_{14}. But then $\{z_1, y_3, y_{24}, y_{34}\}$ is a 4-cut separating $\mathcal{B}$ from the rest of G, and the resulting separation violates Observation 4. This finishes the proof of the lemma. $\square$

Lemma 3.2.10 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6. Let G have a largest clique X of size 4, such that there is no vertex $y \in Y$ with $\deg_X(y) = 3$. Then G has a minor isomorphic to K_6.*

Proof: By the hypothesis of the lemma, and by Observation 1 applied to each edge in X, it follows that Y consists precisely of the six vertices y_{ij} for $1 \leq i < j \leq 4$, where y_{ij} is a common neighbor of x_i, x_j. Applying Observation 1 to x_1y_{12} and x_2y_{12}, it follows that the degree of y_{12} in $G[Y]$ is at least 2. By symmetry, the minimum degree in $G[Y]$ is at least two. Also, the maximum degree in $G[Y]$ is clearly at most 4.

First, suppose that no vertex in $y_{ij} \in Y$ is such that it shares two common neighbors in Y with each of x_i and x_j. (In particular, every vertex in Y has a neighbor outside $X \cup Y$.) By Observation 1 applied to x_1y_{1i} for $i = 2, 3, 4$, y_{12}, y_{13}, y_{14} span a connected subgraph. Hence, without loss of generality, we may assume that y_{12} is adjacent to y_{13} and y_{14}. Further, $G[y_{12}, y_{23}, y_{24}]$ consists of exactly two edges: $y_{12}y_{23}, y_{23}y_{24}$ or $y_{12}y_{24}, y_{24}y_{23}$. By symmetry, we may assume the first outcome. Now since $G[y_{13}, y_{23}, y_{34}]$ is connected, y_{13} must be adjacent to y_{34}. Finally, y_{24} is adjacent to either y_{14} or y_{34} (or both). If the former is true, then the branch sets $\{y_{14}, y_{24}, y_{23}\}, \{y_{12}, y_{13}, y_{34}\}, \{x_1\}, \ldots, \{x_4\}$ witness a K_6 minor in G. If the latter is true, then the branch sets $\{y_{13}, y_{34}, y_{24}\}, \{y_{23}, y_{12}, y_{14}\}, \{x_1\}, \ldots, \{x_4\}$ witness a K_6 minor in G.

Thus we may assume, without loss of generality, that y_{12} shares two common neighbors in Y with each of x_1 and x_2, that is, it is adjacent to y_{13},y_{14},y_{23},y_{24}. If there is no vertex outside $X \cup Y$, then $G[Y]$ must be 4-regular, and thus has all possible edges except those of the form $y_{ij}y_{kl}$ with distinct i, j, k, l. But then $\{y_{12}, y_{13}, y_{34}\}$, $\{y_{14}, y_{24}, y_{23}\}$, and the four vertices in X form the branch sets of a K_6 minor. Thus we may assume that $(X \cup Y, V(G) \backslash (X \cup \{y_{12}\}))$ is a non-trivial separation. But contracting the edges $x_1y_{14}, x_2y_{23}, x_3y_{34}, x_4y_{24}, y_{12}y_{13}$ makes it violate Observation 4. This finishes the proof of the lemma. □

3.3 When G has Clique Number 3

First, we analyze the possible isomorphism classes of $G[N(x)]$, where $N(x)$ is the set of neighbors of a vertex x.

Lemma 3.3.1 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let the size of the largest clique in G be 3. Then for any vertex x, the set $N(x)$ of its neighbors induces a graph isomorphic to one of the graphs $N_0, \ldots, N_5$ shown in Figure 3.7*

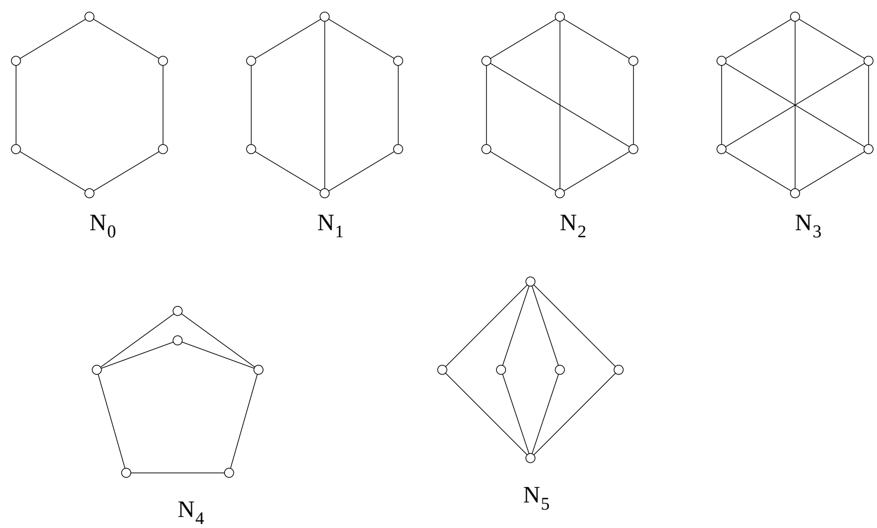

Figure 3.7: Possible neighborhoods of a vertex

Proof: Let N be the graph induced in G by $N(x)$. By Observation 1, the minimum degree in N is two, and hence it has a cycle. N has no triangles, since G has no clique of size 4. Let C be a longest cycle in N. If C has length 6, then the only other edges in N can be the main diagonals of C. Thus N is isomorphic to one of $N_0, \ldots, N_3$. If C has length 5, then the vertex outside C is adjacent to an independent set in C. Thus that vertex is adjacent to exactly two vertices in C, and we get the graph N_4. Finally, suppose C has length 4. The two vertices outside C cannot be adjacent, otherwise N would contains a cycle of length greater than 4. It now follows that N must in fact be isomorphic to N_5 (which is the complete bipartite graph $K_{2,4}$). □

For a graph N, we say that G is *N-free* if no vertex has a neighborhood that induces a graph isomorphic to N.

For the following lemma, refer to Chapters 4 and 5 for the definitions of internal 5-connectivity, graph embedding and representativity (of an embedding).

Lemma 3.3.2 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let the size of the largest clique in G be 3. Then G has a minor isomorphic to K_6.*

Proof: By Lemma 3.3.1, the graphs induced by vertex neighborhoods are isomorphic to one of the graphs $N_0, \ldots, N_5$ shown in Figure 3.7. Suppose there exists a vertex x such that $N(x)$ is isomorphic to N_5. Then by Lemma 3.3.3, we are done. Thus G is N_5-free.

Now applying Lemma 3.3.4, it follows that G is N_i-free for $i = 1, 2, 3, 5$. Finally, applying Lemma 3.3.11, we conclude that all neighborhoods in G are induced 6-cycles. It now follows that G has a embedding in a surface Σ such that each face (region of $\Sigma - G$) is homeomorphic to the open disk and is bounded by a triangle in G. By Euler's formula, it now follows that the Euler genus of Σ is two. Thus G is a triangulation of either the torus or the Klein bottle.

First, let Σ be the torus. We claim that G is internally 5-connected. Suppose not, and let G have a separation (A, B) of order ≤ 4. Then it follows that one of $G[A], G[B]$ (say, $G[A]$) has a planar drawing with the vertices of $A \cap B$ on the exterior face. A simple application of Euler's formula now implies that $A - B$ is empty (and the order of (A, B) is ≤ 3). This proves our claim. By Corollary 5.1.3, G has a K_6 minor.

Finally, let Σ be the Klein bottle. Since G is a triangulation of Σ, with the neighborhoods of each vertex being an induced 6-cycle, it easily follows that the representativity of the embedding must be ≥ 4. From Theorem 5.2.3, it follows that G has a K_6 minor. $\square$

The rest of this section is devoted to the subsidiary lemmas referenced in the above proof.

Lemma 3.3.3 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and let the size of the largest clique in G be 3. Let $x_1 \in V(G)$ be such that the set $N(x_1)$ of its* *neighbors induces a graph isomorphic to the graph N_5 shown in Figure 3.7. Then G has a minor isomorphic to K_6.*

Proof: First, notice that among $N_0, \ldots, N_5$, all but N_5 have maximum degree ≤ 3.

Let $N(x_1) = \{x_2, x_3, y_1, \ldots, y_4\}$ such that x_2 and y_1 are the two vertices of degree 4 in $G[N(x_1)]$. Since x_1 has degree 4 in $N(x_2)$, $N(x_2)$ must also be isomorphic to N_5. Now, the other vertex of degree 4 in $N(x_2)$ cannot be y_1, since G has no clique of size 4. Hence there must be a neighbor y_5 of x_2 that is adjacent to x_3, y_2, y_3, y_4. Now x_2 has degree 4 in $N(y_5)$, and hence $N(y_5)$ is also isomorphic to N_5. The other vertex of degree 4 in $N(y_5)$ (besides x_2) cannot be x_1. We claim that it cannot be y_1 either. Suppose the contrary. Then, $N(x_3)$ has the 4-cycle $y_1 x_1 x_2 x_5 y_1$. However, since $N(x_3)$ cannot contain y_2, y_3 or y_4, this 4-cycle is a connected component of $G[N(x_3)]$, which is a contradiction. This proves the claim.

Thus y_5 has a neighbor y_6 (distinct from y_1) such that it is adjacent to x_3, y_2, y_3, y_4. But now y_5 has degree 4 in $N(y_6)$ (which is thus isomorphic to N_5). The other vertex y_7 of degree 4 in $N(y_6)$ cannot be identical to x_2 since g has no clique of size 4. It cannot be identical to x_1 since G is 6-regular. By an argument similar to the one used for the above claim, y_7 is not identical to y_1 either. Thus $G[N(x_3)]$ contains a path $y_1 x_1 x_2 y_5 y_6 y_7$ of length 6. It then follows that $G[N(x_3)]$ contains a 6-cycle (and that y_7 is adjacent to y_1). But then, G has a K_6 minor with branch sets given by $\{x_1\}, \{x_2\}, \{x_3\}, \{y_2, y_5\}, \{y_3, y_6\}, \{y_4, y_1, y_7\}$. □

Lemma 3.3.4 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_5-free, and $x_1 \in V(G)$ be such that the set $N(x_1)$ of its neighbors induces a graph isomorphic to one of the graphs N_1, N_2, N_3 shown in Figure 3.7. Then G has a minor isomorphic to K_6.*

Proof: The neighborhood of x_1 consists of a 6-cycle with at least one main diagonal. Let the main diagonal be the edge $x_2 x_3$ and the 6-cycle be $y_1 x_2 y_1' y_3' x_3 y_3 y_1$. (Note that the other two diagonal edges, $y_1 y_3'$ and $y_3 y_1'$, may or may not exist.) By Observation 1 and 6-regularity respectively, x_2 and x_3 share at least one, and at most two, common neighbors besides x_1.

Suppose that x_2 and x_3 share two common neighbors y_2 and y_2'. Notice that in $G[N(x_2)]$, x_1 and x_3 are adjacent vertices of degree exactly three. Applying Lemma 3.3.1 to x_2, it follows that there must be a perfect matching between $\{y_1, y_1'\}$ and $\{y_2, y_2'\}$. Using a similar argument for x_3, it follows that there must be a perfect matching between $\{y_3, y_3'\}$ and $\{y_2, y_2'\}$. Without loss of generality, we may assume the edges y_1y_2 and $y_1'y_2'$. Now if the second perfect matching between $\{y_3, y_3'\}$ and $\{y_2, y_2'\}$ has the edges y_2y_3' and $y_2'y_3$, then G has a K_6 minor: contract the edges $y_1y_3, y_2y_3', y_1'y_2'$. Thus the perfect matching has the edges y_2y_3 and $y_2'y_3'$. Thus G has a rooted subgraph isomorphic to the graph H_7 shown in Figure 3.8.

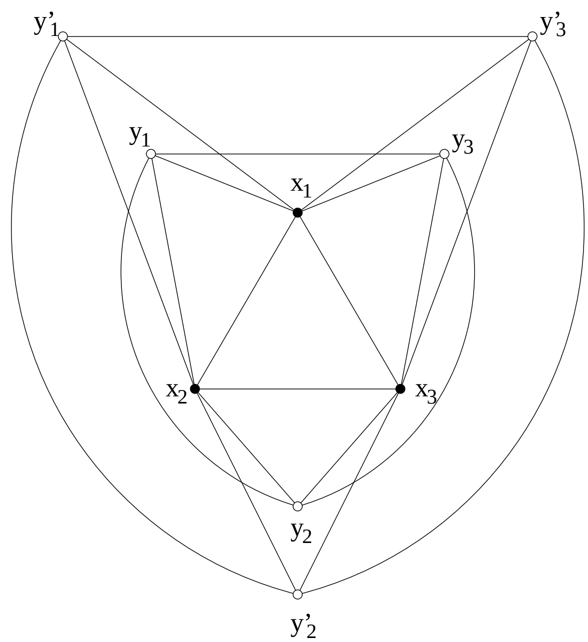

Figure 3.8: Graph H_7

If there are two disjoint edges between $\{y_1, y_2, y_3\}$ and $\{y_4, y_5, y_6\}$, then there is a 6-cycle passing through those 6 vertices, and contracting three independent edges in that cycle yields a K_6 minor in G. Hence, without loss of generality, we may assume that the only edges between $\{y_1, y_2, y_3\}$ and $\{y_4, y_5, y_6\}$ are, possibly, y_1y_2' and y_1y_3'. (Note that y_1 cannot be adjacent to y_1' since G has no clique of size 4.) But now, contract the edges x_iy_i for $i = 1, 2, 3$ in G to get a graph G'. The only vertex in G' with degree less than 6 is possibly y_3', and G' has at least 7 vertices because $V(G) \neq X \cup Y$ (where $X = \{x_1, x_2, x_3\}$

and $Y = \{y_1, y_2, y_3, y'_1, y'_2, y'_3\}$). Thus $G' \in \mathcal{F}_6$, contradicting the minimality of G.

Thus x_2 and x_3 share exactly one common neighbor besides x_1, and let y_3 be that common neighbor. Let y_4 and y_5 be the sixth neighbors, respectively, of x_2 and x_3. Notice that in $G[N(x_2)]$, x_1 has degree exactly 3. Applying Lemma 3.3.1 to x_2, we get one of the following outcomes, by symmetry:

1. G contains the edges y_1y_3, y_3y_4, y'_1y_4 ($G[N(x_2)]$ is isomorphic to the graph N_4)

2. G contains the edges y_1y_3, y_1y_4, y'_1y_4 ($G[N(x_2)]$ is isomorphic to one of the graphs N_1, N_2, N_3)

3. G contains the edges y_3y_4, y_1y_4, y'_1y_4 ($G[N(x_2)]$ is isomorphic to the graph N_4)

Applying Lemma 3.3.1 to x_3, we additionally get one of the following outcomes:

A. G contains the edges y_2y_3, y_3y_5, y'_2y_5 or the edges y'_2y_3, y_3y_5, y_2y_5 ($G[N(x_3)]$ is isomorphic to the graph N_4)

B. G contains the edges y_2y_3, y_2y_5, y'_2y_5 or the edges y'_2y_3, y_2y_5, y'_2y_5 ($G[N(x_3)]$ is isomorphic to one of the graphs N_1, N_2, N_3)

C. G contains the edges y_3y_5, y_2y_5, y'_2y_5 ($G[N(x_3)]$ is isomorphic to the graph N_4)

Note that for x_2, in outcomes 1 and 2, there are two realizations each that are symmetric. However, for x_3, in the corresponding outcomes A and B, that symmetry is broken, and we need to consider both realizations. Note also that in outcomes 1 and 3 (respectively, A and C), there are no more edges in $G[N(x_2)]$ (respectively, $G[N(x_3)]$) than the ones indicated.

By symmetry, the combinations of two outcomes that can happen above are: 1 and A, 1 and B, 1 and C, 2 and B, 2 and C, and 3 and C. In these cases, G has a rooted subgraph isomorphic, respectively, to the graphs H_8 or H'_8, H_9 or H'_9, H_{10}, H_{11} or H'_{11}, H_{13}, H_{14} shown in Figures 3.9, 3.10, 3.11, 3.12, 3.14, 3.15 respectively. By applying Lemmas 3.3.5, 3.3.6, 3.3.7, 3.3.8, 3.3.9 and 3.3.10 in turn, it follows that G has a K_6 minor. This finishes the proof of the lemma. $\square$

Lemma 3.3.5 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to one of the graphs H_8 or H_8' shown in Figure 3.9. Then G has a minor isomorphic to K_6.*

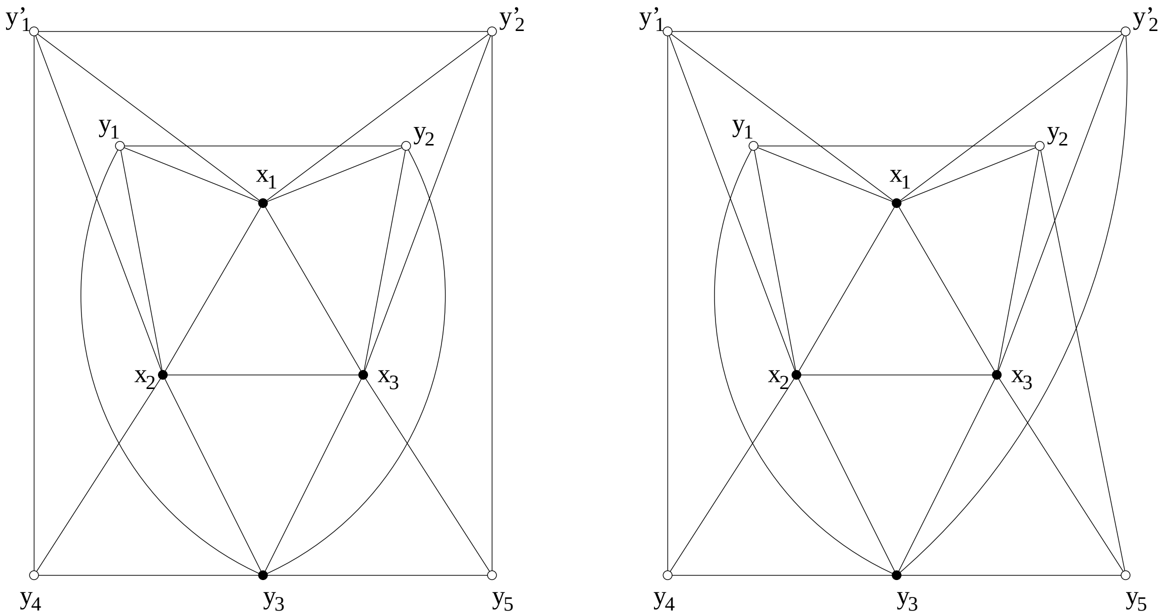

Figure 3.9: Graphs H_8 and H_8'

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. Suppose that G has a rooted subgraph isomorphic to H_8. If either of the edges y_1y_5 or y_2y_4 exist, then G has a K_6 minor. Now y_4 must have a neighbor outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$). Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. If $\mathcal{B}$ attaches at y_2 as well, or at y_1 and y_5, then G has a K_6 minor as before. Hence the possible attachments of $\mathcal{B}$ are restricted to y_1', y_2', y_4 and at most one of y_1, y_5. In any case, the resulting separation in G violates Observation 4.

Now suppose G has a rooted subgraph isomorphic to H_8'. Note that if either of the edges $y_1'y_5$ or y_2y_4 exist, then G has a K_6 minor. The rest of the proof then proceeds as for graph H_8 above, and this finishes the proof of the lemma. $\square$

Lemma 3.3.6 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to one of the graphs H_9 or H_9' shown in Figure 3.10. Then G has a minor isomorphic to K_6.*

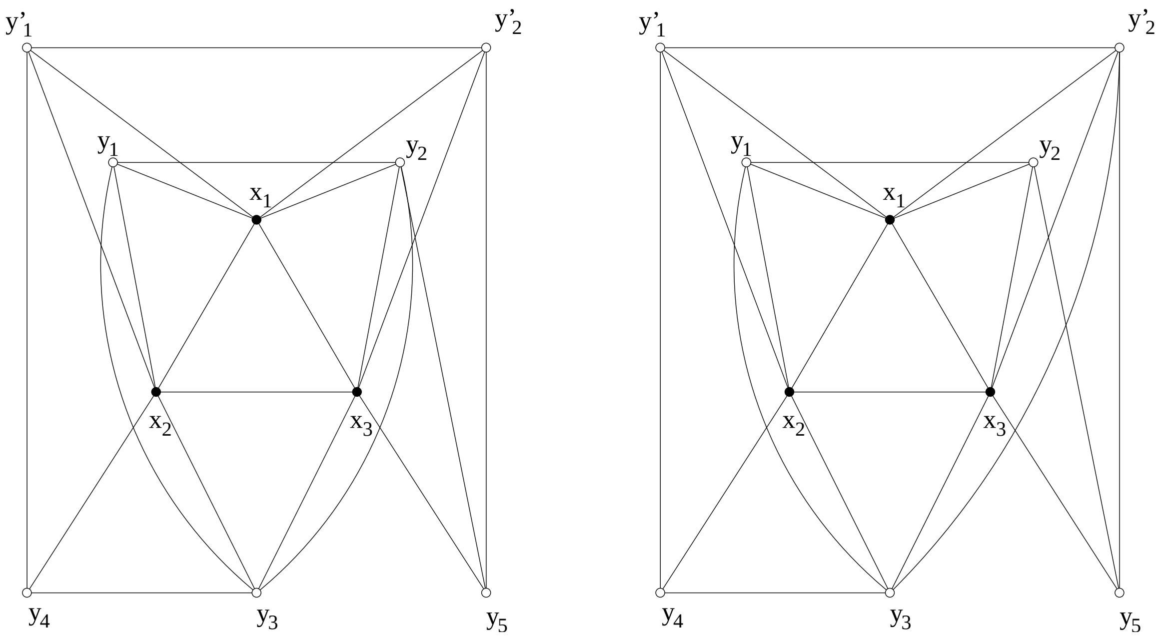

Figure 3.10: Graphs H_9 and H_9'

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. Suppose first that G has a rooted subgraph isomorphic to H_9. If the edge y_1y_5 exists, then G has a K_6 minor. Otherwise, it must have a neighbor outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$). Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. Now $\mathcal{B}$ does not attach at y_5, otherwise G has a K_6 minor as before. If $\mathcal{B}$ attaches at y_2, then y_2 has its sixth neighbor in $\mathcal{B}$. From Lemma 3.3.1 applied to y_2, and the fact that G is N_5-free, it then follows that this neighbor must also be adjacent to y_5, but then G has a K_6 minor. Thus $\mathcal{B}$ cannot attach at y_2. The possible attachments of $\mathcal{B}$ are now restricted to y_1, y_1', y_4 and at most one vertex among y_2', y_3. (If $\mathcal{B}$ attaches at both y_2' and y_3, then G has a K_6 minor witnessed by the branch sets $\{x_1, y_1', y_4\}, \{x_2, x_3\}, \{y_2, y_5\}, \{y_2'\} \cup V(\mathcal{B}), \{y_1\}, \{y_3\}$.) In either case, the resulting separation in G violates Observation 4.

Thus we may assume that G has a rooted subgraph isomorphic to H_9'. If y_4 is adjacent to either y_2 or y_5, then G has a K_6 minor. Since G has no clique of size 4, y_4 is not adjacent to y_1 either. Thus y_4 has a neighbor outside $X \cup Y$. Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. Clearly, $\mathcal{B}$ does not attach at y_2 or y_5, otherwise G has a K_6 minor as above.

We claim that $\mathcal{B}$ cannot attach at y_2'. Suppose it does. Then y_2' has its sixth neighbor z in $\mathcal{B}$. Also, note that y_1' is not adjacent to y_3 since G has no clique of size 4. Now from Lemma 3.3.1 applied to y_2', it follows that G either has the edge $y_1'y_5$ or the edges $y_1'z, zy_5$. In either case, G has a K_6 minor. This proves the claim. But then the possible attachments of $\mathcal{B}$ are restricted to y_1, y_1', y_3, y_4, and the resulting separation in G violates Observation 4. This proves the lemma. □

Lemma 3.3.7 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to the graph H_{10} shown in Figure 3.11. Then G has a minor isomorphic to K_6.*

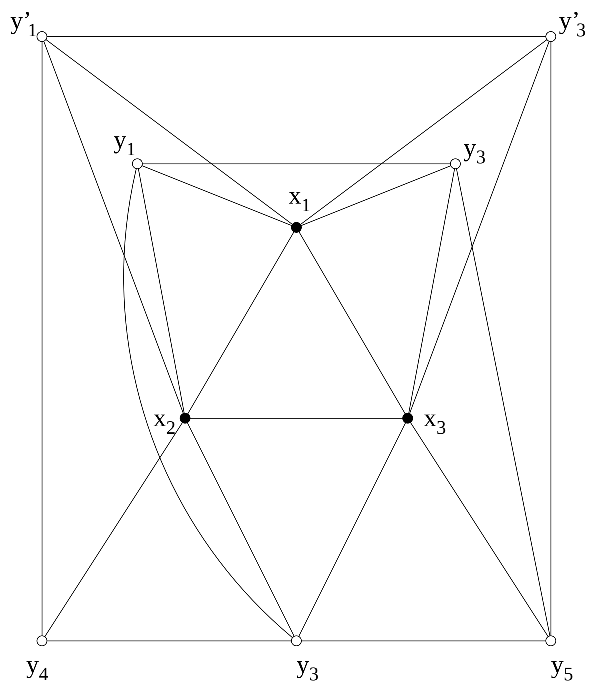

Figure 3.11: Graph H_{10}

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. If y_2 is adjacent to y_4, then G has aK_6 minor. Also, if y_2 is adjacent to y_1', then G has a K_6 minor, witnessed by the branch sets $\{x_1\}, \{x_2, x_3\}, \{y_1, y_3, y_4\}, \{y_1'\}, \{y_2\}, \{y_2', y_5\}$. Now y_2 has a neighbor outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$). Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. If $\mathcal{B}$ attaches at y_4 or y_1', then G has a K_6 minor as before. Thus the possible attachments of $\mathcal{B}$ are now restricted to $\{y_1, y_2, y_2', y_3, y_5\}$.

But then the attachments of $\mathcal{B}$ separate it from the rest of G, and that separation violates Observation 4. (To get the K_5 needed for the hypothesis of Observation 4, choose the branch sets $\{y_1, x_2, x_3\}, \{y_2, x_1\}, \{y_5\}, \{y_3, y_4, y_1'\}, \{y_2'\}$.) This contradiction finishes the proof of the lemma. $\square$

Lemma 3.3.8 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to one of the graphs H_{11} or H_{11}' shown in Figure 3.12. Then G has a minor isomorphic to K_6.*

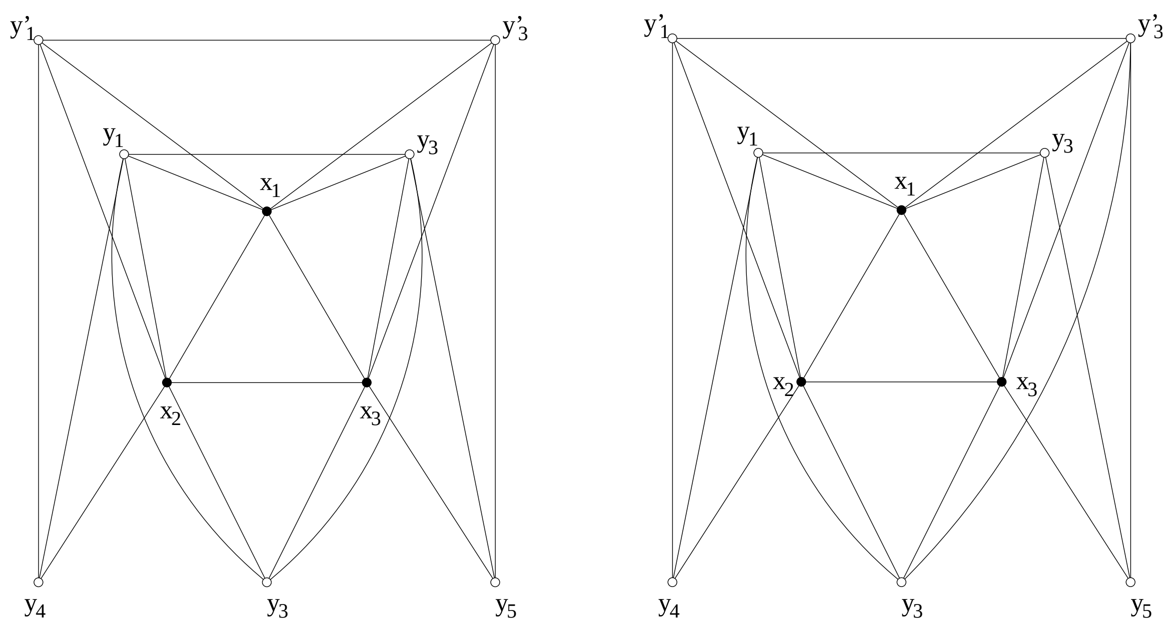

Figure 3.12: Graphs H_{11} and H_{11}'

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. First, suppose that G has a rooted subgraph isomorphic to H_{11}. Applying Lemma 3.3.1 to y_1, it follows that exactly one of the following outcomes occur:

1. the vertex y_2', or

2. the vertex y_5, or

3. a vertex z outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$)

By symmetry, applying Lemma 3.3.1 to y_2 yields similar results. Now if y_1 is adjacent to y_2' (outcome 1 above), then y_2' must be adjacent to y_4 (from Lemma 3.3.1). Looking at the five neighbors of y_2' uncovered so far, and applying Lemma 3.3.1 to it, we conclude that y_5 is adjacent to either y_1' or y_4. In either case, G has a K_6 minor.

Now suppose y_1 is adjacent to y_5. From Lemma 3.3.1 applied to y_1, the edge y_4y_5 exists. Also, we may assume that y_2 is not adjacent to y_4 or y_1', otherwise G has a K_6 minor. Thus y_2 has it sixth neighbor z outside $X \cup Y$. Applying Lemma 3.3.1 to y_2, it follows that z is adjacent to both y_3 and y_5. But then, applying Lemma 3.3.1 to y_3 now yields a contradiction.

Thus the only possibility left is that outcome 1 above occurs for y_1, and its symmetrical version occurs for y_2. Thus they both have neighbors z_1, z_2 respectively outside $X \cup Y$ (where z_1, z_2 may be identical). If $z_1 = z_2 = z$, then from Lemma 3.3.1 applied in turn to y_1 and y_2, it follows that z is adjacent to y_4 and y_5. Note that y_3 cannot be adjacent to y_4, y_5 or z since G has no clique of size 4. Applying Lemma 3.3.1 to y_3, it now follows that the two additional neighbors of y_3 must be y_1' and y_2' (and that $G[N(y_3)]$ is isomorphic to N_1). But then G has a K_6 minor.

Thus z_1, z_2 are distinct vertices. Now applying Lemma 3.3.1 to y_1 and y_2, we get: z_1 is adjacent to y_3 and y_4, and z_1 is adjacent to y_3 and y_5. Further, applying Lemma 3.3.1 to y_3, it follows that z_1, z_2 are adjacent. Thus G has a rooted subgraph isomorphic to the graph H_{12} shown in Figure 3.13.

Let $P = X \cup \{y_1, y_2, y_3\}$. Consider the graph $G_0 = G \setminus P$, with $\Omega = y_1', y_4, z_1, z_2, y_5, y_2'$ (in that cyclic order). Since G_0 has $\frac{1}{2}(6(n-6)+24) = 3n-6$ edges (where n is the number of its vertices), it cannot have a planar drawing with $V(\Omega)$ on the boundary of the infinite face.

We claim that G_0 has no $\leq$3-separation (A, B) with $V(\Omega) \subseteq A$. Suppose it does; choose (A, B) as above such that its order k is minimal. Then by Menger's theorem, there exist k vertex disjoint paths from $V(\Omega)$ to $A \cap B$. It now follows easily that the separation $(A \cup P, B)$ in G violates Observation 4. This proves our claim.

Thus by Theorem 3.1.2 the society (G_0, Ω) has a cross. It is now easy to check that G has a K_6 minor: let $V(\Omega)$ and the cross form four branch sets of the K_6, and let the other two branch sets be $\{x_1, y_1, y_2\}$ and $\{x_2, x_3, y_3\}$.

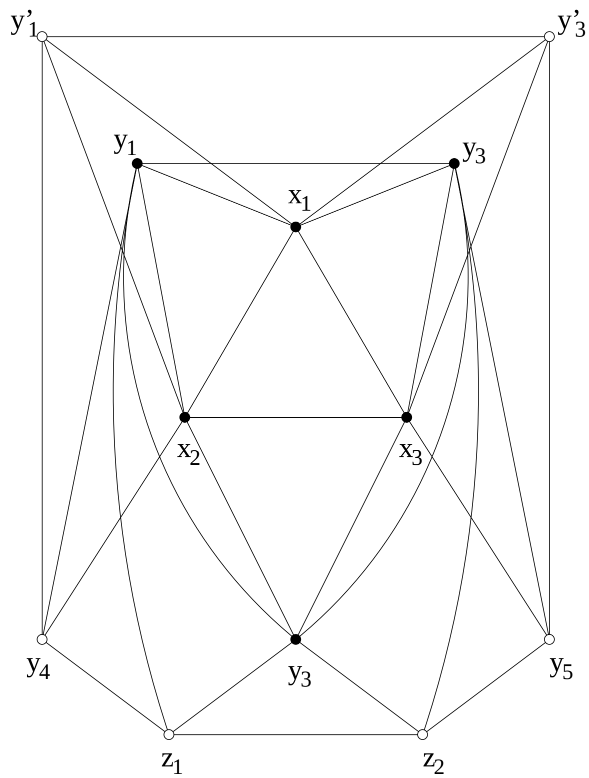

Figure 3.13: Graph H_{12}

Finally, suppose G has a rooted subgraph isomorphic to the graph H'_{11}. Note that if y_5 is adjacent to either y'_1 or y_4, then G has aK_6 minor. Thus y_5 has a neighbor outside $X \cup Y$. Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. Clearly, $\mathcal{B}$ does not attach at y'_1 or y_4, otherwise G has a K_6 minor as before. If $\mathcal{B}$ attaches at y_1, then y_1 has its sixth neighbor z in $\mathcal{B}$. Note that z is not adjacent to y_4, since $\mathcal{B}$ does not attach at y_4. Applying Lemma 3.3.1 to y_1, it now follows that G has the edge y_2y_4. But then G has a K_6 minor. Thus we may assume that $\mathcal{B}$ does not attach at y_1. The possible attachments of $\mathcal{B}$ are now restricted to y_2, y'_2, y_3, y_5, and the resulting separation in G violates Observation 4. This finishes the proof of the lemma. $\square$

Lemma 3.3.9 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to the graph H_{13} shown in Figure 3.14. Then G has a minor isomorphic to K_6.*

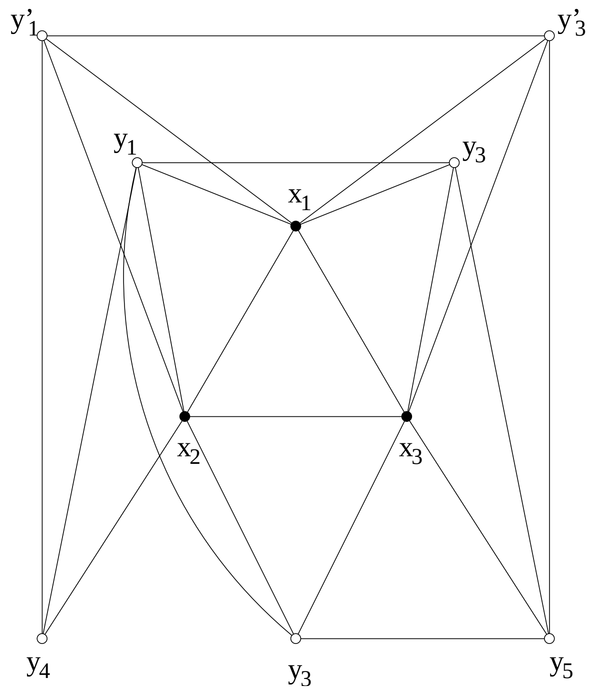

Figure 3.14: Graph H_{13}

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. If y_2 is adjacent to either y_1' or y_4, then G has a K_6 minor. Thus y_2 has a neighbor outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$). Let $\mathcal{B}$ be an $(X \cup Y)$-bridge of G containing such a neighbor. Clearly, $\mathcal{B}$ does not attach at y_1' or y_4, otherwise G has a K_6 minor as before.

We claim that $\mathcal{B}$ does not attach at y_1. Suppose it does; then y_1 has a sixth neighbor z outside $X \cup Y$. Note that z is not adjacent to y_4 since $\mathcal{B}$does not attach at y_4. Also, y_4 is not adjacent to y_3 since G has no clique of size 4. But then, applying Lemma 3.3.1 to y_1, y_4 is adjacent to y_2, and G has a K_6 minor. This proves the claim.

Now the possible attachments of $\mathcal{B}$ are restricted to y_2, y_2', y_3, y_5, and the resulting separation in G violates Observation 4. This proves the lemma. $\square$

Lemma 3.3.10 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, such that it is N_5-free and has a rooted subgraph isomorphic to the graph H_{14} shown in Figure 3.15. Then G has a minor isomorphic to K_6.*

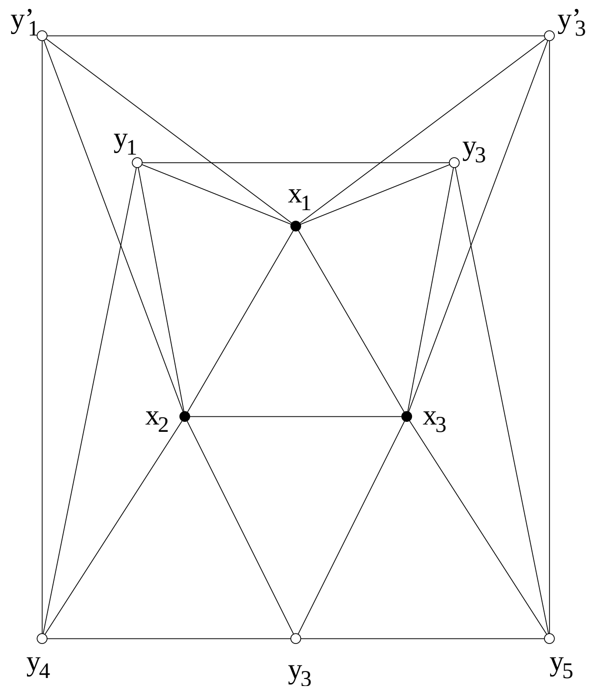

Figure 3.15: Graph H_{14}

Proof: By Lemma 3.2.1, we may assume that G has no clique of size 4. If either of the edges y_1y_2' or y_2y_1' exist, then G has a K_6 minor. Similarly, we may assume there is no $(X \cup Y)$-bridge of G that attaches at y_1, y_2' or at y_2, y_1'.

Now y_1 has a neighbor outside $X \cup Y$ (where $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_1', y_2, y_2', y_3, y_4, y_5\}$). Let $\mathcal{B}_1$ be an $(X \cup Y)$-bridge of G containing such a neighbor. If $\mathcal{B}_1$ does not attach at y_5, then its attachments are restricted either to y_1, y_2, y_3, y_4 or y_1, y_1', y_3, y_4. In either case, the resulting separation in G violates Observation 4. Thus $\mathcal{B}_1$ must attach at y_5.

Using a similar argument as in the previous paragraph, we get an $(X \cup Y)$-bridge $\mathcal{B}_2$ of G that attaches at y_2 and y_4. If $\mathcal{B}_1 \neq \mathcal{B}_2$, then G has a K_6 minor. Thus $\mathcal{B}_1 = \mathcal{B}_2 = \mathcal{B}$. If $\mathcal{B}$ does not attach at y_3, then its only attachments are y_1, y_2, y_4, y_5, and the resulting separation in G violates Observation 4. Thus $\mathcal{B}$ attaches precisely at $y_1, \ldots, y_5$.

Interchanging the roles of y_1, y_2 with y_1', y_2', and using the above argument, we further get an $(X \cup Y)$-bridge $\mathcal{B}'$ that attaches precisely at $y_1', y_2', y_3, y_4, y_5$. By 6-regularity, y_4 and y_5 are each incident to exactly one edge of $\mathcal{B}$ and $\mathcal{B}'$ respectively. But now, it follows that the number of edges in $\mathcal{B}$ that are incident to exactly one attachment is five, an odd number.

This contradicts the 6-regularity of G, and finishes the proof of the lemma. □

We now turn towards eliminating N_4 from the list of graphs induced by the neighborhood of a vertex in G. For the subsequent discussion, G is a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, with no clique of size 4, and which is N_i-free for $i = 1, 2, 3, 5$.

For an edge uv in G, the vertices u and v share either two or three common neighbors. By "a common neighbor of the edge e", we mean a common neighbor of its endpoints. We shall call an edge *thick* if it has three common neighbors. The following observations are trivial.

Observation 5 *If uv is a thick edge, then $G[N(u)]$ and $G[N(v)]$ are both isomorphic to N_4. Also, v is one of the two vertices of degree three in $G[N(u)]$ (and vice versa).*

Observation 6 *The thick edges in G induce a 2-regular subgraph (unless there are no thick edges at all).*

Observation 7 *No triangle in G has more than one thick edge. In particular, the graph induced by the thick edges consists of cycles of length at least 4.*

Observation 8 *If $N(x)$ contains a 4-cycle, for some vertex x, then the two thick edges incident with x join it to two alternate vertices on the 4-cycle.*

On inspecting N_4, it follows that two adjacent thick edges must share exactly two of their common neighbors. (In other words, if x_1x_2 and x_2x_3 are both thick edges, then there are exactly two vertices y that are incident to x_1, x_2, x_3.) Now let e be a thick edges and e', e'' be the two thick edges incident to it. It now follows that the number of common neighbors that e, e', e'' share is either one or two. If they share two common neighbors, then we call e a *regular* edge. Otherwise (that is, if they share exactly one common neighbor), we call e an *irregular* edge.

Lemma 3.3.11 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and $x \in V(G)$ be such that the set $N(x)$ of its neighbors induces a graph isomorphic to the graph N_4 shown in Figure 3.7. Then G has a minor isomorphic to K_6.*

Proof: By the hypothesis of the lemma, G has a thick edge. (In fact, since the thick edges form a 2-regular subgraph, G has at least 3 thick edges.)

Consider two adjacent thick edges. If they are both regular, then by Lemma 3.3.13, G has a K_6 minor. If exactly one of them is regular, then we are done by Lemma 3.3.14. Finally, if they are both irregular, then we are done by Lemma 3.3.18. □

Lemma 3.3.12 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$. Let x_1x_2, x_2x_3, x_3x_4 be three thick edges (with $x_1 \neq x_4$). Thus the graph induced by $P = \{x_2, x_3\} \cup N(x_2) \cup N(x_3)$ is isomorphic to one of those shown in Figure 3.16 (depending on whether x_2x_3 is a regular edge.) Then x_4 does not share any neighbor with x_2 other than the ones it shares with x_3 (that is, other than y_1, y_2). By symmetry, x_1, x_3 also have no common neighbors other than y_1, y_2.*

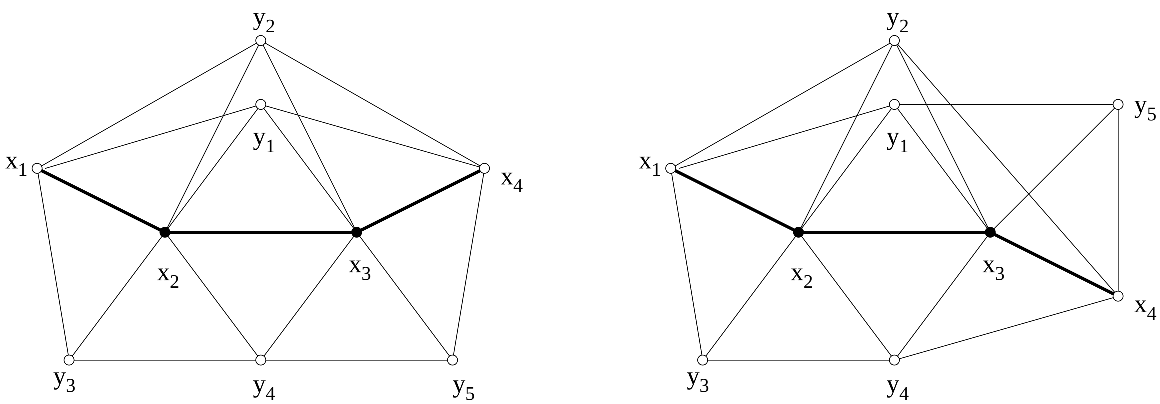

Figure 3.16: Neighborhood of a thick edge

Proof: Clearly, y_1 and y_2 have no more edges in $G[P]$ than those indicated, since G has no cliques of size 4.

First, suppose x_2x_3 is irregular, that is, $G[P]$ is the graph on the right in Figure 3.16. If the conclusion of the lemma were false, then x_4 must be adjacent to either x_1 or y_3. But then $N(y_1)$ or $N(y_4)$ (respectively) has a 4-cycle. By Observation 8, one of the edges y_1x_2, y_1x_3 (respectively, y_4x_2, y_4x_3) must be thick, which contradicts Observation 6.

Thus, we may assume that x_2x_3 is regular, that is, $G[P]$ is the graph on the left in Figure 3.16. Suppose that the conclusion of the lemma is false, and that x_4 is adjacent to

x_1. Since $G[N(y_1)]$ contains a 4-cycle. This leads to a similar contradiction as above.

Now suppose x_4 is adjacent to y_3. Since y_3 is not adjacent to y_1, y_2 or x_3, it follows that x_4y_3 has exactly two common neighbors. Thus the edge x_4y_3 cannot be thick, and y_3 is adjacent to y_5. But then $G[N(y_4)]$ contains a 4-cycle, and neither y_4x_2 nor y_4x_3 is a thick edge, which is a contradiction as above. □

Lemma 3.3.13 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and let G have two adjacent thick edges that are both regular. Then G has a K_6 minor.*

Proof: From Lemma 3.3.12, it follows that G has a rooted subgraph isomorphic to the graph shown in Figure 3.17. (The edges x_2x_3, x_3x_4 are the two adjacent regular edges.) The sixth

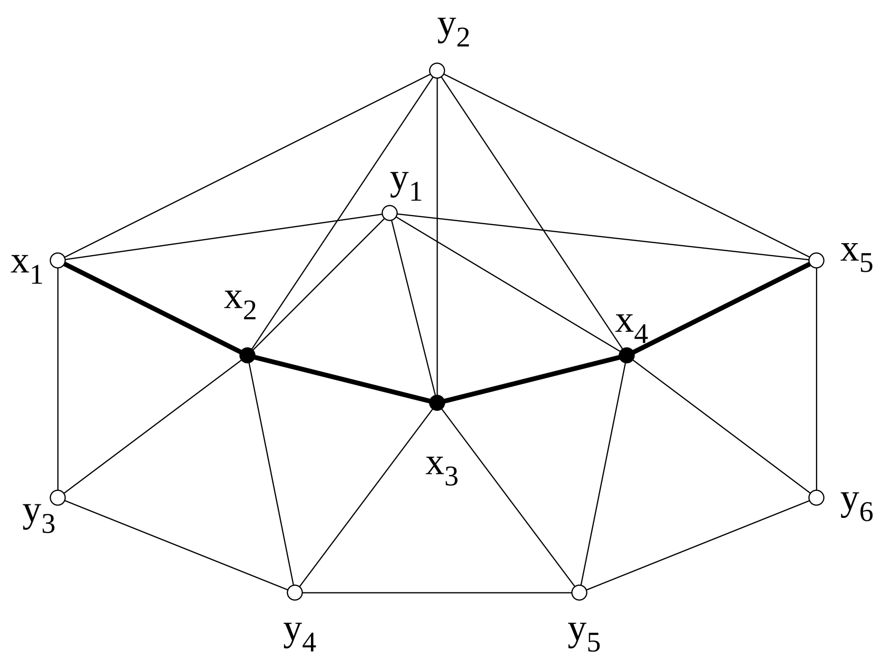

Figure 3.17: Graph with two adjacent regular edges

neighbor of y_1 cannot be in $\{y_2, \ldots, y_6\}$. Also, $N(y_1)$ must be isomorphic to N_0 (a 6-cycle). Thus y_1 has a neighbor z_1 adjacent to x_1 and x_5. Similarly, y_2 has a neighbor z_2 adjacent to x_1 and x_5.

If z_1, z_2 are distinct, then one of the edges x_5z_1, x_5z_2 must be thick. (Note that by Observation 7, none of the edges x_5y_1, x_5y_2, x_5y_6 can be thick.) We may assume, without loss of generality, that x_5z_1 is thick. Since $N(x_5)$ is isomorphic to N_4, it follows that z_1, z_2 are adjacent. But then $N(z_2)$ has a 4-cycle $x_1y_2x_5z_1x_1$. But neither of the edges z_2y_2, z_2x_5 are thick, which contradicts Observation 8. Thus $z_1 = z_2$, and call this vertex x_6.

Now $N(x_6)$ has a 4-cycle $x_1y_1x_5y_2$, and by Observation 8, it follows that the edges x_6x_1, x_6x_5 are thick. Thus G has a rooted subgraph isomorphic to the graph H_{15} shown in Figure 3.18.

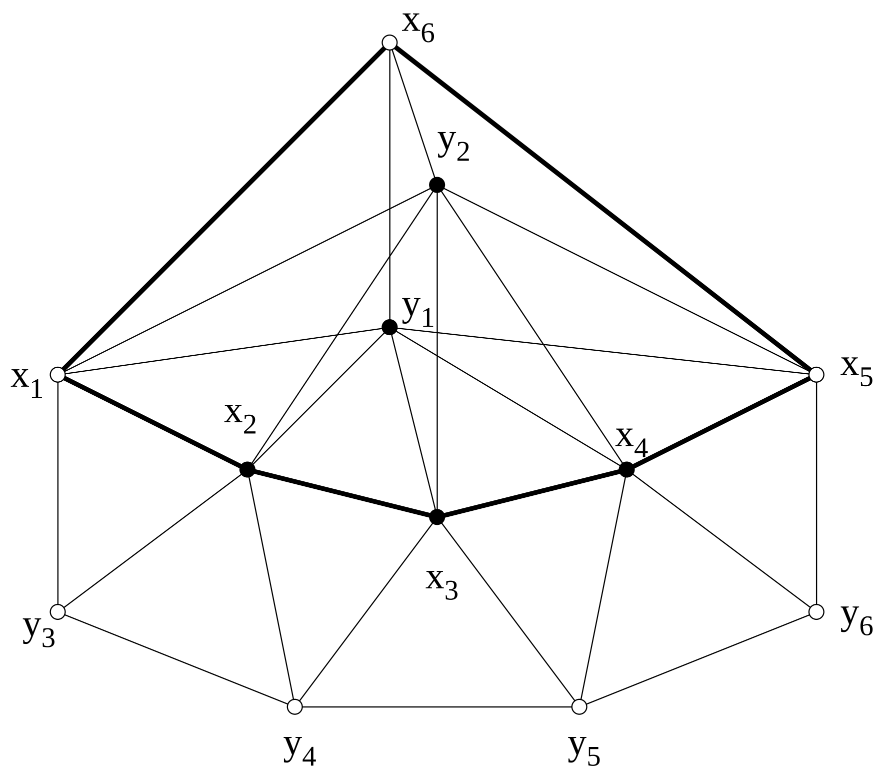

Figure 3.18: Graph H_{15}

By examining $N(x_1), N(x_5)$ respectively, it follows that x_6 cannot be adjacent to y_3 or y_6. Now $N(x_6)$ has two more vertices other than the ones shown, and x_1, x_5 must each be adjacent to one of them. Neither of those vertices can be y_4 or y_5, since by Lemma 3.3.12, the edges x_1y_5, x_5y_4 cannot exist. Thus x_6 has neighbors y_7, y_8 distinct from all the vertices in H_{15}, such that x_1 is adjacent to y_7 and x_5 is adjacent to y_8. Also, by examining $N(x_1)$, $N(x_5)$ and $N(x_6)$ in turn, we get the edges y_7y_3, y_7y_8 and y_8y_6. But then, contracting a perfect matching between $\{x_1, \ldots, x_6\}$ and $\{y_3, \ldots, y_8\}$ yields a graph in $\mathcal{F}_6$, a contradiction. This proves the lemma. □

Lemma 3.3.14 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and let G have two adjacent thick edges x_2x_3, x_3x_4, such that x_2x_3 is regular and x_3x_4 is irregular. Then G has a K_6 minor.*

Proof: By Lemma 3.3.12, without loss of generality, G has a rooted subgraph isomorphic to the graph H_{16} shown in Figure 3.19.

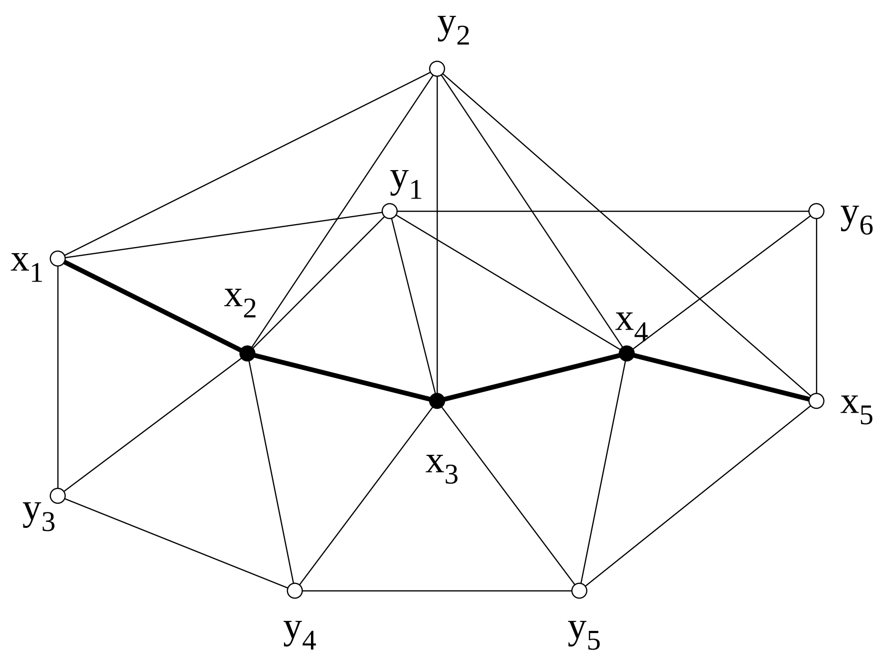

Figure 3.19: Graph H_{16}

First, suppose x_4x_5 is regular. Let the second thick edge incident with x_5 join it to a vertex x_6. We claim that x_6 cannot be any vertex in H_{16}. (By Lemma 3.3.12, it suffices to show that it is not x_1 or y_3.) Since $N(x_5)$ is isomorphic to N_4, it follows that x_6 is adjacent to at least one vertex among y_2, y_5. This means that $x_6 \neq y_3$. (Note that x_3 cannot be adjacent to y_5, otherwise $N(y_4)$ violates Observation 8.) If $x_6 = x_1$, on the other hand, $N(y_2)$ has a 5-cycle (and hence is isomorphic to N_5). But none of the edges y_2x_i for $i = 2, 3, 4$ can be thick, which is a contradiction. This proves the claim that x_6 is outside $V(H_{16})$.

Now since x_4x_5 is regular, x_6 shares three neighbors with x_5, comprised of: (i) the vertex y_6, (ii) exactly one vertex among y_2, y_5 and (iii) a vertex y_7. It follows by an argument similar to the above claim that y_7 cannot be in $V(H_{16}) \cup \{x_6\}$.

Suppose now that x_6 is adjacent to y_6, y_5, y_7. From $N(x_5)$, we get the edges x_5y_7, y_2y_7. Now $N(y_2)$ cannot be isomorphic to N_4. Hence we get the edge x_1y_7. Thus G has a rooted subgraph isomorphic to the graph H_{17} shown in Figure 3.20, and by Lemma 3.3.15, we are done.

On the other hand, if x_6 is adjacent to y_6, y_2, y_7, then $N(y_2)$ must be a 6-cycle, and we get the edge x_6x_1. Thus G has a rooted subgraph isomorphic to the graph H_{18} shown in Figure 3.21, and by Lemma 3.3.16, we are done.

Finally, suppose x_4x_5 is irregular. Then G has a rooted subgraph isomorphic to the graph H_{19} shown in Figure 3.22, and by Lemma 3.3.17, we are done. □

Lemma 3.3.15 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and let G have a rooted subgraph isomorphic to the graph H_{17} shown in Figure 3.20. Then G has a K_6 minor.*

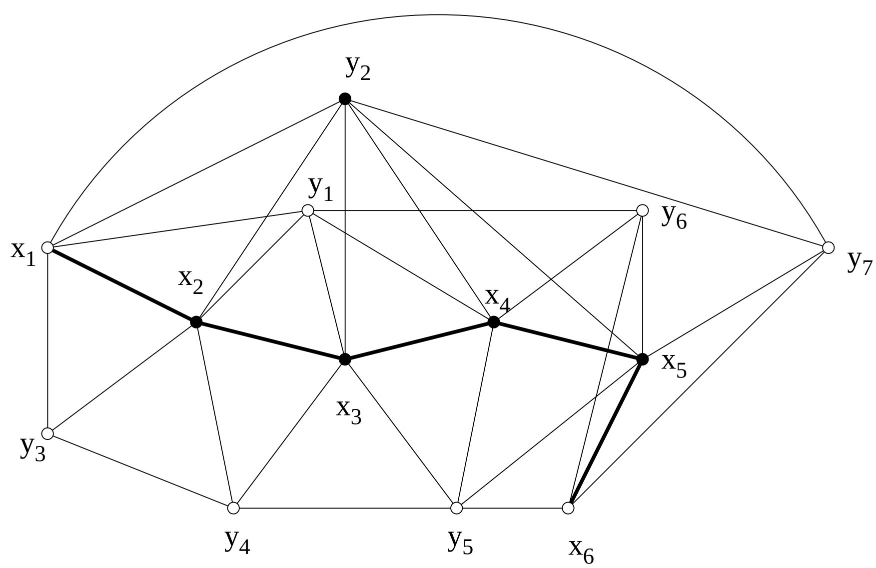

Figure 3.20: Graph H_{17}

Proof: $N(y_1)$ cannot be isomorphic to N_4. Let y_8 be the sixth neighbor of y_1. Thus y_8 is adjacent to x_1 and y_6. Now y_8 is either x_6 or y_7 or a vertex outside $V(H_{17})$.

If $y_8 = x_6$, then we get the edge x_1x_6. Now $N(x_1)$ is isomorphic to N_4 and the second thick edge incident to it (that is, other than x_1x_2) is either x_1x_6 or x_1y_7. But then we respectively get the edges x_6y_3 or y_7y_3. In either case, G has a K_6 minor.

If $y_8 = y_7$, then from $N(y_1)$ we get the edge y_6y_7. But G has no clique of size 4, so this is a contradiction. Thus y_8 is outside $V(H_{17})$ (and is adjacent to x_1 and y_6). Again, consider $N(x_1)$, which is isomorphic to N_4. The second thick edge incident to it (that is, other than x_1x_2) is either x_1x_7 or x_1y_8. From $N(y_1)$ we get the edges y_7y_3 or y_8y_3 respectively in the first and second cases. In either case, G has a K_6 minor, which proves the lemma. □

Lemma 3.3.16 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and let G have a rooted subgraph isomorphic to the graph H_{18} shown in Figure 3.21. Then G has a K_6 minor.*

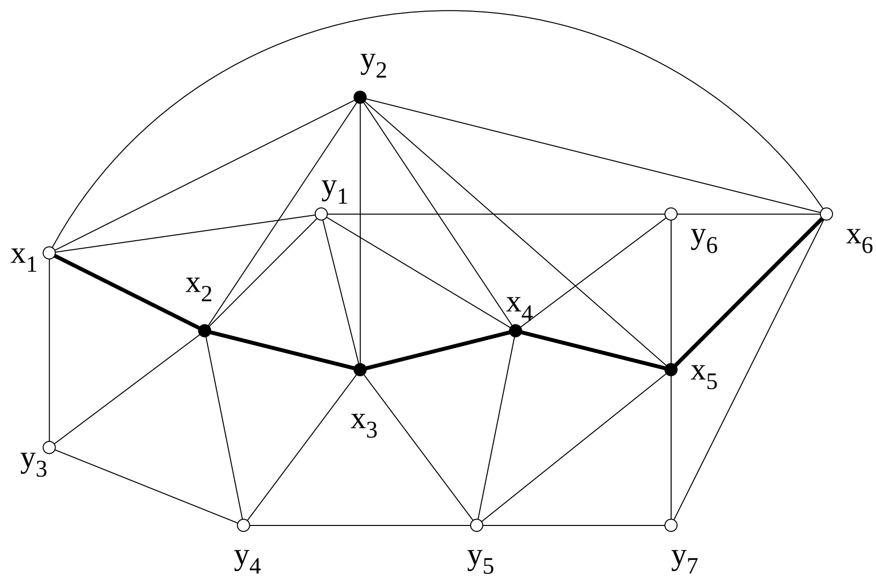

Figure 3.21: Graph H_{18}

Proof: The vertex y_1 has a neighbor y_8 other than the five shown in Figure 3.21. By Lemma 3.3.12, y_8 is either x_6 or y_7 or is outside $V(H_{18})$.

If $y_8 = x_6$, then consider $N(x_6)$, which is isomorphic to N_4. Now the edge x_6y_1 is not thick, as $N(y_1)$ induces a 6-cycle. Thus x_6x_1 must be thick. But then x_1x_2, x_2x_3 are adjacent regular edges. By Lemma 3.3.13, we are done.

If $y_8 = y_7$, then from $N(y_1)$ we get the edges y_7x_1, y_7y_6. But then G has a K_6 minor.

Finally, suppose that y_8 is outside $V(H_{18})$. From $N(y_1)$ we get the edges y_8x_1, y_8y_6. Consider $N(x_1)$, which is isomorphic to N_4. Exactly one of the edges x_1x_6, x_1y_8 is thick.

If x_1x_6 is thick, then from $N(x_1)$, we get the edges x_6y_3, x_6y_8. But then x_6 has degree at least seven, a contradiction.

Thus x_1y_8 is thick. Then from $N(x_1)$, we get the edges y_8x_6, y_8y_3. But then G has a K_6 minor, which proves the lemma. □

Lemma 3.3.17 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and let G have a rooted subgraph isomorphic to the graph H_{19} shown in Figure 3.22. Then G has a K_6 minor.*

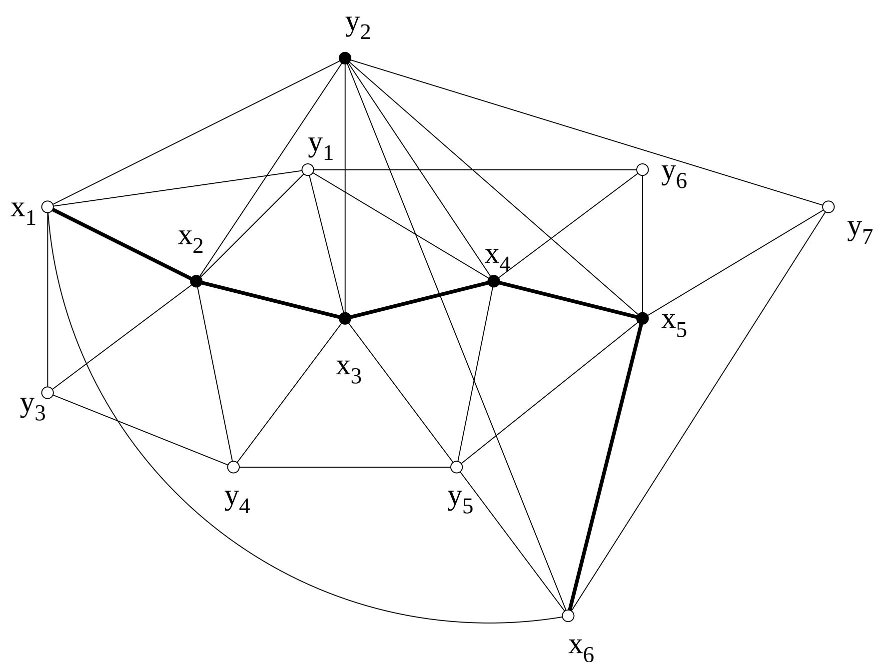

Figure 3.22: Graph H_{19}

Proof: $N(y_5)$ cannot be N_4, and hence y_5 a neighbor y_8 that is adjacent to y_4, x_6. By Lemma 3.3.12, y_8 must be outside $V(H_{19})$.

Now $N(x_6)$ is isomorphic to N_4, and exactly one of the edges x_6x_1, x_6y_8 is thick. If x_6x_1 is thick, then from $N(x_6)$ we get the edges x_1y_7, x_1y_8. But then x_1 has degree at least 7, a contradiction.

Thus the edge x_6y_8 is thick. But then, from $N(x_6)$ we get the edge y_8y_7, and G has a K_6 minor. This proves the lemma. □

Lemma 3.3.18 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and such that all thick edges in G are irregular. Then G has a K_6 minor.*

Proof: G has a rooted subgraph isomorphic to the graph on the right of Figure 3.16. By Lemma 3.3.12, x_4 has neighbors x_5 and y_6 besides the ones shown in the figure, where x_4x_5 is a thick edge. Since the edge x_3x_4 is also irregular by the hypothesis of the lemma, it follows that x_5 is adjacent to: (i) y_5 and y_6, and (ii) exactly one vertex among y_4, y_2.

First, suppose x_5 is adjacent to y_5, y_6, y_4. Now $N(y_4)$ must induce a 6-cycle, and the sixth neighbor of y_4 is outside $\{x_1, \ldots, x_5, y_1, \ldots, y_6\}$. This neighbor y_7 is adjacent to y_3 and x_5. Thus G has a rooted subgraph isomorphic to the graph H_{20} shown in Figure 3.23, and by Lemma 3.3.19, we are done.

Now suppose that x_5 is adjacent to y_5, y_6, y_2. From $N(x_4)$ we also get the edge y_4y_6. Now $N(y_4)$ must induce a 6-cycle, and thus y_4 must have a neighbor y_7 outside $\{x_1, \ldots, x_5, y_1, \ldots, y_6\}$, such that y_7 is adjacent to y_3 and y_6. Thus G has a rooted subgraph isomorphic to the graph H_{22} shown in Figure 3.25, and by Lemma 3.3.20, we are done. This proves the lemma. $\square$

Lemma 3.3.19 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and such that all thick edges in G are irregular. Let G have a rooted subgraph isomorphic to the graph H_{20} shown in Figure 3.23. Then G has a K_6 minor.*

Proof: $N(y_2)$ must induce a 6-cycle. We claim that the sixth neighbor of y_2, other than the five shown in the figure, must be outside $V(H_{20})$. Suppose not; then that vertex must be y_7, since G has no clique of size 4.

From $N(y_2)$ we now get the edges y_7y_6, y_7x_1, and it follows that $N(y_7)$ also induces a 6-cycle. In particular, the edge y_7x_5 is not thick. But $N(x_5)$ contains a 4-cycle $x_4y_6y_7y_4$, and by Observation 8 x_5y_7 must be thick, which is a contradiction. This proves the claim.

Thus y_2 has a neighbor y_8 outside $V(H_{20})$, such that y_8 is adjacent to x_1 and y_6. Now if x_5, x_1 are adjacent, then they can share at most one neighbor, by six-regularity. This is a

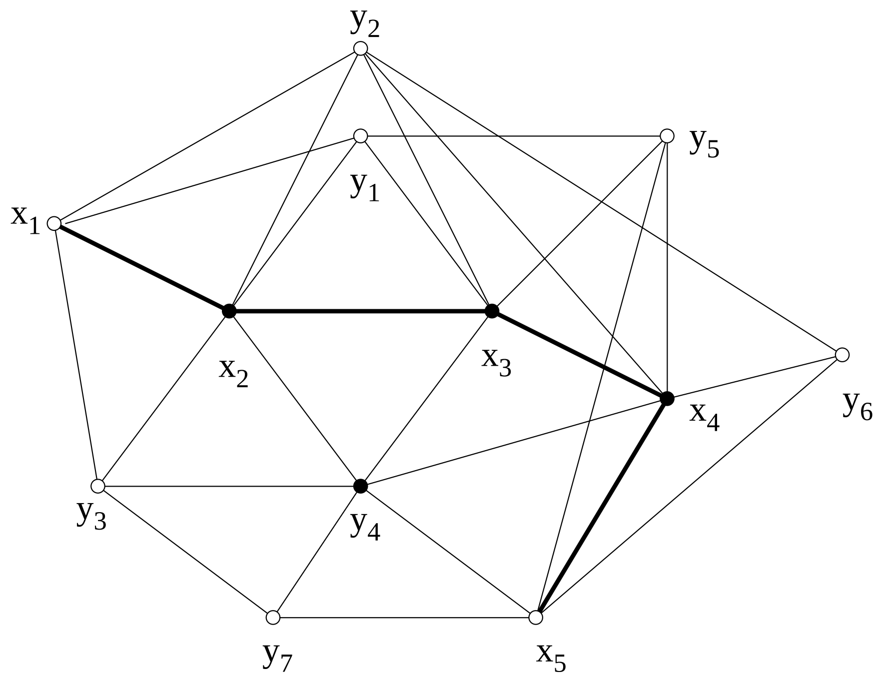

Figure 3.23: Graph H_{20}

contradiction, so x_5 cannot be adjacent to x_1. Thus, by Lemma 3.3.12, the sixth neighbor z of x_5 (other than those shown in Figure 3.23) is either y_8 or is a vertex outside $V(H_{20})$.

Suppose $z = y_8$. Then by examining $N(x_5)$ (and noting that x_4x_5 is regular), we get the edges y_8y_5, y_8y_7. In particular, the edge x_5y_8 must be thick. Thus G has a rooted subgraph isomorphic to the graph H_{21} shown in Figure 3.24. Since $N(y_2)$ induces a 6-cycle, the edge y_8y_2 is not thick. Hence y_8x_1 must be a thick edge, and from $N(y_8)$ we deduce that x_1 must be adjacent to y_7 and (exactly) one vertex among y_5, y_6. This means x_1 has degree at least 7, a contradiction.

Finally, suppose the vertex z is outside $V(H_{20})$. From $N(x_5)$ we deduce that z must be adjacent to y_7 and y_8. But then G has a K_6 minor, which proves the lemma. □

Lemma 3.3.20 *Let G be a minor-minimal graph in $\mathcal{F}_6$, with minimum degree ≥ 6, and with no clique of size 4. Let G be N_i-free for $i = 1, 2, 3, 5$, and such that all thick edges in G are irregular. Let G have a rooted subgraph isomorphic to the graph H_{22} shown in Figure 3.25. Then G has a K_6 minor.*

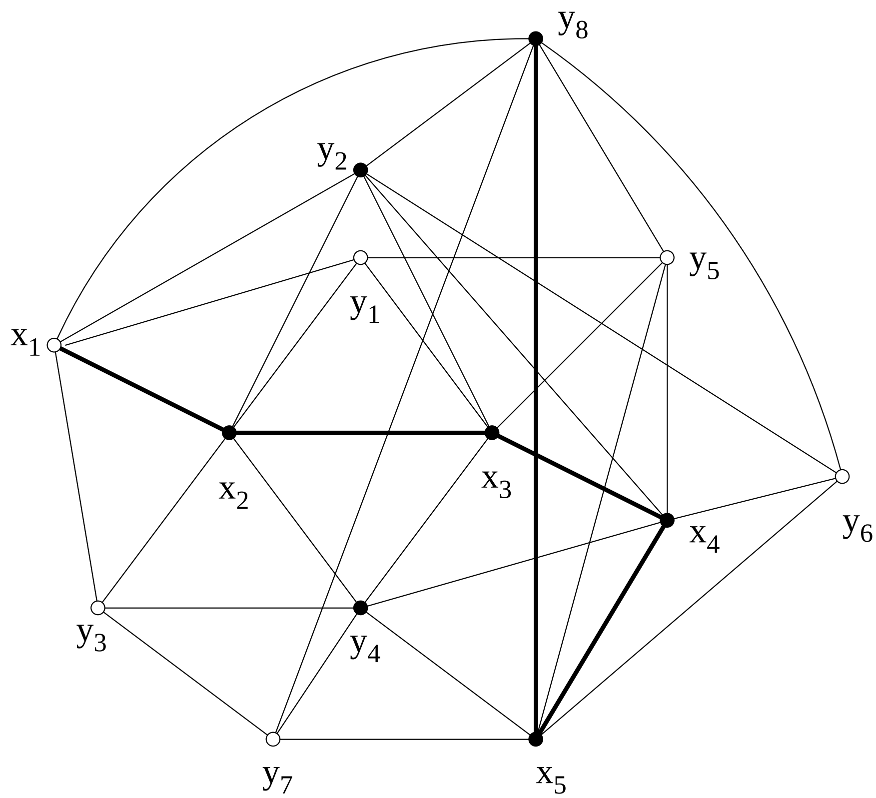

Figure 3.24: Graph H_{21}

Proof: $N(y_2)$ must induce a 6-cycle. We claim that the sixth neighbor of y_2, other than the five shown in the figure, must be outside $V(H_{22})$. Suppose not; then that vertex must be y_7, which must then be adjacent to x_1 and x_5. But then $N(y_6)$ contains a 4-cycle that violates Observation 8. This proves the claim.

Thus y_2 has a neighbor y_8 outside $V(H_{22})$, such that y_8 is adjacent to x_1 and x_5. Thus G has a rooted subgraph isomorphic to the graph H_{23} shown in Figure 3.26.

Note that the graph H_{23} has left-right symmetry as shown in the figure. Now $N(x_1)$ induces a graph isomorphic to N_4. Let z_1 be the sixth neighbor of x_1 (other than the five shown in the figure). From $N(x_1)$, and the fact that x_1x_2 is irregular, we deduce the edges z_1x_1, z_1y_8, z_1y_1. We claim that z_1 must be outside $V(H_{23})$. Suppose not; then by Lemma 3.3.12, z_1 must be identical to x_5, y_6 or y_7. However, $z_1 \neq x_5$ since G has no clique of size 4. Also, $z_1 \neq y_7$, otherwise $N(y_3)$ has a 4-cycle that violates Observation 8. Finally, $z_1 \neq y_6$, otherwise G has a K_6 minor. This proves our claim that z_1 is outside $V(H_{23})$.

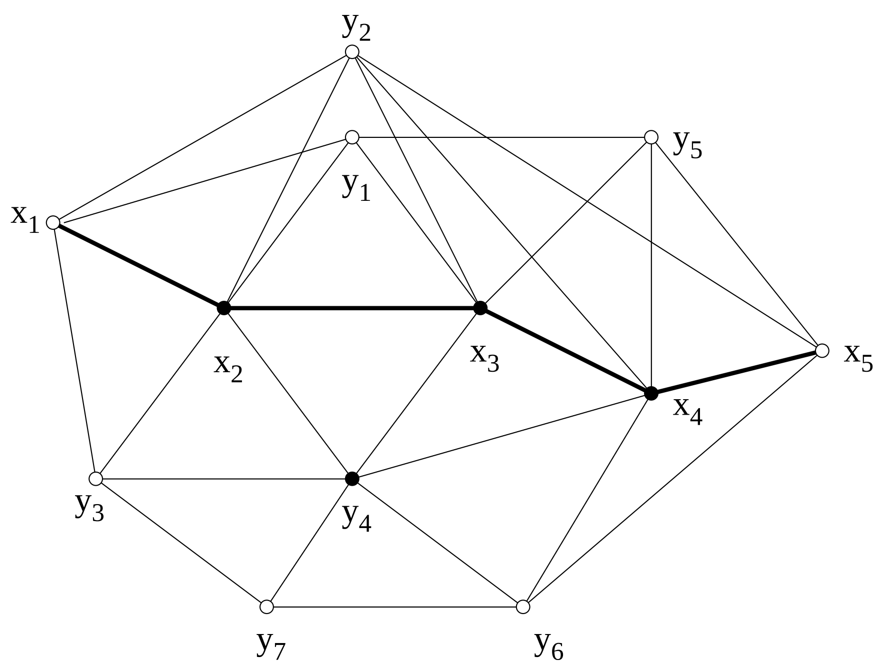

Figure 3.25: Graph H_{22}

By symmetry, x_5 also has a neighbor z_2 outside $V(H_{23})$, such that z_2 is adjacent to x_5, y_8 and y_5. (Note that z_1, z_2 may be identical.) First, suppose that z_1, z_2 are distinct. If the edge y_8x_1 is thick, then from $N(x_1)$ it follows that y_8 is adjacent to y_3. Similarly, if y_8x_5 is thick, then y_8 must be adjacent to y_6. Since $\deg(y_8) = 6$, it follows that at least one of the edges y_8x_1, y_8x_5 (say, the former) is *not* thick. From $N(x_1)$, we now deduce the edge z_1y_3. But then G has a K_6 minor.

Thus, we may assume that $z_1 = z_2 = z$. Now $N(y_8)$ has a 4-cycle $x_1y_2x_5zx_1$. Since $N(y_2)$ induces a 6-cycle, the edge y_8y_2 is not thick. Thus by Observation 8, both the edges y_8x_1 and y_8x_5 must be thick. From $N(x_1), N(x_5)$ respectively, we now deduce the edges y_8y_3, y_8y_6. But then, from $N(y_8)$, it follows that y_3, y_6 are adjacent. This contradicts the fact that G has a no clique of size 4, and thus proves the lemma. $\square$

3.4 Proof of Main Theorem

We now prove Theorem 3.1.1. Let G be as in the statement of the theorem. If G is not six-regular, then deleting an edge incident to a vertex of degree ≥ 7 creates at most one low-degree vertex, and hence the resulting graph is also in $\mathcal{F}_6$, contradicting the minimality

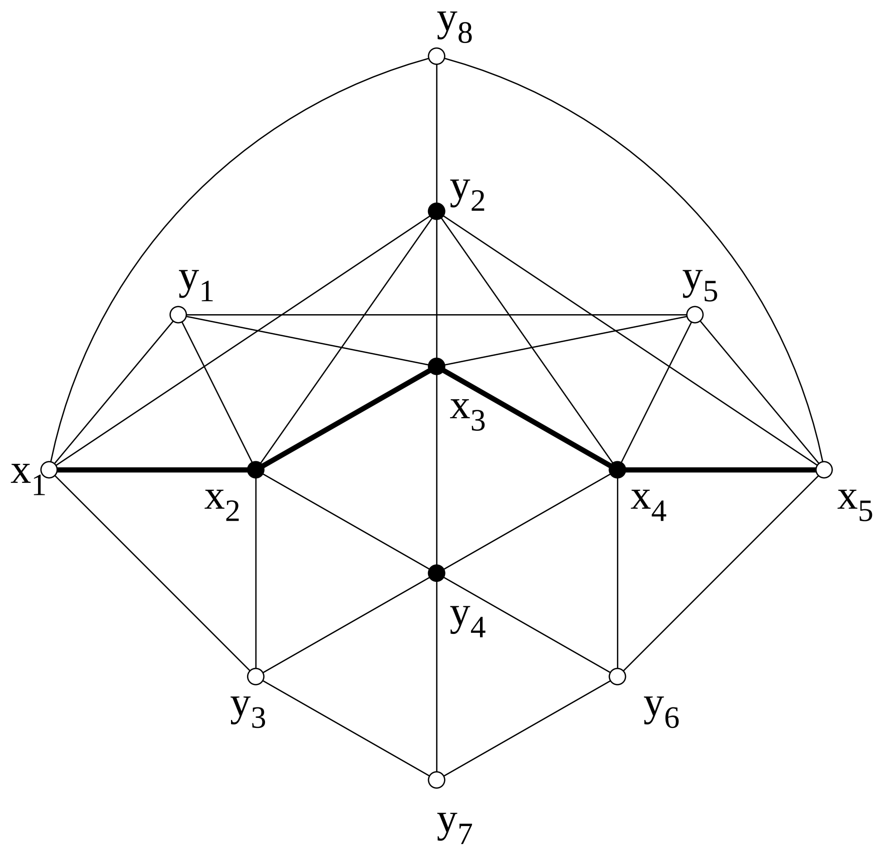

Figure 3.26: Graph H_{23}

of G. Hence it follows that G is *6-regular*. Also, we may assume that the size of a maximum clique is ≥ 3 by Observation 1, and ≤ 5 (otherwise Theorem 3.1.1 holds trivially). The theorem now follows by applying lemmas 3.1.3, 3.2.1 and 3.3.2. □

3.5 Jorgensen's Conjecture in Surfaces of Non-negative Euler Characteristic

In the proof of Lemma 3.3.2, the cases when G is a triangulation of the torus or the Klein bottle also follow respectively from some recent work of Fijavž [13, 14], which is based on results by Altshuler [1], Negami [37, 36], and Thomassen [55]. More precisely, [13] proves that there are precisely 4 minor-minimal 6-regular graphs in the torus (K_7, $K_8 - 4K_2$, $K_9 - C_9$, and $K_9 - 3K_3$). It is easy to see that each of the 4 graphs has a K_6 minor. Similarly, [14] proves that there are precisely 3 (explicitly given) minor-minimal 6-regular graphs in the Klein bottle, and it is easy to check that each of them has a K_6 minor.

By Euler's formula, a 6-connected graph in the torus or the Klein bottle must be 6-regular (and is hence a triangulation). From the above discussion, it now follows that Jorgensen's conjecture (Conjecture 1.2.1) is true for graphs embedded in the torus or the Klein bottle.

Also, by Euler's formula, no graph embedded in the projective plane is 6-connected; thus Jorgensen's conjecture holds vacuously for projective-planar graphs. More interestingly, however, the corresponding statement for 5-connected graphs is also true: Fijavž and Mohar [15] proved that a 5-connected graph in the projective plane has a K_6 minor if and only if it has representativity at least 3. It is easy to see that a graph embedded in the projective plane with representativity ≤ 2 is apex, and hence a 5-connected graph in the projective plane that has no K_6 minor must be apex.

CHAPTER IV

A SPLITTER THEOREM FOR SURFACE TRIANGULATIONS

4.1 Introduction

We begin with some basic notation. For a motivation of the splitter theorem proved in this chapter, refer to Section 1.2.3. Graphs in this chapter are allowed to have parallel edges or loops.

By a *graph embedding* G, we mean a graph embedded in a surface (compact 2-dimensional manifold) Σ, such that every face (that is, region of $\Sigma - G$) is homeomorphic to an open disk. (Such an embedding is sometimes called a *cellular* embedding.)

Given a graph H embedded in a surface Σ, performing a sequence of contractions and deletions of edges yields a graph embedding G in Σ that is called a *surface minor* of H. We follow the usual convention for edge contractions in graph embeddings — when contracting an edge e that lies in a triangular face, one of the resulting parallel edges created by the contraction is deleted. (Note that parallel edges that are created by non-facial triangles are *not* deleted by the operation.)

A closed curve in a surface Σ is called *contractible* if it is homotopic to a point. Otherwise, it is called *non-contractible* or *essential.* Thus a simple closed curve is contractible if and only if it bounds an open disk. Given a graph embedding G and a closed curve C in Σ, the G*-degree* of C (or simply degree, when the embedding is understood) is defined to be $|C \cap G|$.

Let G be a graph embedded in a surface Σ, and C be a contractible simple closed curve in Σ that does not meet the interior of any edge of G. (In other words, $C \cap G \subseteq V(G)$.) Let R_1 and R_2 be the closures of the two regions of $\Sigma - C$, such that R_1 is a closed disk. (Then R_2 is not a closed disk, unless Σ is the sphere S^2.) Let A, B be the intersections of $V(G)$

with R_1, R_2 respectively. The resulting separation (A, B) of G is called a *local separation*, and C is called a *boundary curve* for that separation.

If (A, B) is a local separation in H, then fixing a boundary curve for the separation and then performing an edge contraction or deletion yields a corresponding local separation in the new graph. Thus if G is obtained from H by a sequence of edge contractions and deletions, then a local separation in H uniquely defines a local separation in G.

We now define the notion of connectivity for our splitter theorem. An embedding G in a surface Σ is *locally 5-connected* if

1. G has at least six vertices,
2. for every local $\leq$3-separation (A, B), $A - B$ is empty (in particular, (A, B) is trivial), and
3. for every local 4-separation (A, B), $A - B$ has at most one vertex.

Local 5-connectivity is weaker than 5-connectivity, in that it allows vertices of degree 4, and vertex cuts (of any size) that lie on non-contractible closed curves in Σ.

An embedding G in a surface Σ is called a *triangulation* if every face (that is, region of $\Sigma \setminus G$) is bounded by a triangle of G. Thus a triangulation cannot have loops. It can have parallel edges, however. (A pair of parallel edges would always trace a non-contractible curve.)

By "triangulation", we sometimes refer to the underlying graph, if no ambiguity arises. Thus, we say that a triangulation has a K_6 minor when the underlying graph has a minor isomorphic to K_6.

A triangulation is called *simple* if the underlying graph is simple, that is, has no loops or parallel edges.

For a vertex v in an embedding, the *star of* v denotes the set of edges incident with v. Given an embedding G in a surface Σ, fixing an orientation (say, clockwise) at every vertex v yields a cyclic permutation of the star of v. (This is sometimes referred to as the *rotation* at v, and the collection of rotations is called a *rotation scheme* for G.) A subset of the star of v that is contiguous with respect to the rotation at v is said to be a *fan around* v.

We now define splits, which are the inverses of edge contractions performed on triangulations. (There are more general variants of this operation which are sometimes also called splits (see Chapter 2, for instance) , but the following definition is relevant particularly to triangulations.) Let E_1, E_2 be two fans around v such that they intersect in precisely two edges, and their union is the star of v. Now replace v by two adjacent vertices v_1, v_2, with v_i incident with the edges in E_i for $i = 1, 2$. The resulting triangulation is said to be obtained from the original one by *splitting* the vertex v. Conversely, *contracting* the edge v_1v_2 in the new triangulation (and deleting the resulting parallel edges) gives back the original triangulation.

A triangulation H obtained by applying a sequence of splits to a triangulation G is called an *expansion* of G. For a vertex v in G, the set of vertices in H resulting from v is called the *branch set* of v in H. (Note that the branch sets are determined uniquely by a sequence of splits that leads from G to H. However, such a sequence itself may not be uniquely determined.)

Now we define a stronger version of the split operation that, for simple triangulations, turns out to be precisely the kind that preserves local 5-connectivity. A split as defined in the paragraph above is called a *strong split* if the following conditions are satisfied: (i) E_1, E_2 each have at least 3 edges, and at least one of them has ≥ 4 edges (in particular, v has degree ≥ 5) and (ii) for $i = 1, 2$, if E_i has precisely three edges e_1, e_2, e_3, in that order according to the rotation at v, then the end-point x_2 of e_2 (other than v) must have degree ≥ 5 in G.

We now prove that strong splits are precisely those that preserve local 5-connectivity, as long as the vertex v being split is not incident with any parallel edges. (This technical assumption about v can be eliminated by suitably defining local 5-connectivity and strong splits, but we choose not to do so for simplicity.)

Lemma 4.1.1 *Let G be a locally 5-connected triangulation and v a vertex of G that is not incident with any parallel edges. Let G' be the triangulation obtained by splitting v. Then G' is locally 5-connected if and only if the split is strong.*

Proof: The forward implication is trivial to check: if conditions (i) or (ii) in the definition of a strong split are violated, then G' violates conditions (ii) or (iii) respectively of the definition of local 5-connectivity. (Notice that if v has degree 4 but is incident with two parallel edges tracing a non-contractible curve, then a split may preserve local 5-connectivity without being strong.)

For the converse implication, suppose v_1, v_2 are the vertices created by a strong split of v, such that the resulting triangulation G' is not locally 5-connected. Condition (i) in the definition of a strong split ensures that G' does not violate condition (ii) of local 5-connectivity. Thus it has a local 4-separation (A', B') with $|A' - B'| \geq 2$.

We claim that $B' - A'$ must be disjoint from $\{v_1, v_2\}$. Suppose not; then contracting the edge v_1v_2 back yields a local 4-separation (A, B) in G with $|A - B| \geq 2$. This contradicts the local 5-connectivity of G, which proves our claim.

Suppose $A' - B'$ contains both v_1 and v_2. Since G is locally 5-connected, the resulting local 4-separation (A, B) must be such that $A - B = \{v\}$ (and $A' - B' = \{v_1, v_2\}$). It follows that condition (i) of a strong split must have been violated, which is a contradiction.

Now if $A' - B'$ contains exactly one of v_1, v_2, it is easy to check that one of the conditions in the definition of a strong split must be violated. Thus we may assume that $A' \cap B'$ contains both v_1 and v_2. Since v is not incident with any parallel edges, v_1, v_2 must appear consecutively along any boundary curve for (A', B'). But now, contracting the edge v_1v_2 back gives a local $\leq$3-separation (A, B) with $|A - B| \geq 2$. Again, this contradicts the local 5-connectivity of G, and thus proves the lemma. $\square$

4.2 Preliminaries

First, we make the following easy observations for triangulations:

Observation 9 *If G and H are triangulations of a surface Σ such that G is a surface minor of H, then H is an expansion of G, that is, H can be obtained from G by a sequence of splits alone. Every intermediate embedding in this sequence is also a triangulation.*

Proof: G can be obtained from H by a sequence of edge contractions and deletions. It is easy to see that the contractions and deletions can be done in any order without affecting the final triangulation G in the sequence. We choose to perform all the contractions before doing any edge deletions. Let G' be the graph obtained from H by performing all the contractions. Clearly, G' is also a triangulation, and has the same number of vertices as G. Since G is a subgraph of G', it follows that $G = G'$. In other words, H is an expansion of G. The conclusion now follows easily. $\square$

Observation 10 *If (A, B) is a local separation in a triangulation G, then there is a cycle C in G with $V(C) = A \cap B$, such that the simple closed curve traced by C is homotopic to a boundary curve for (A, B).*

Proof: Let $\widetilde{C}$ be a boundary curve for (A, B). Thus $\widetilde{C} \cap G = A \cap B$. Since G is a triangulation, it follows that $\widetilde{C}$ is homotopic to a cycle C in G with the required properties. $\square$

If the cycle C is as in Observation 10 above, we say that C *induces* the local separation (A, B). Thus the above Observation states that every local separation in a triangulation is induced by some cycle.

Observation 11 *Let (A, B) is a local ≤ 5-separation in a locally 5-connected triangulation H. Then $A - B$ induces a connected subgraph of H. Further, if $|A - B| \geq 2$, then every vertex of $A \cap B$ is adjacent to at least one vertex of $A - B$.*

Proof: By the local 5-connectivity of H, and the fact that $H[A]$ has a planar drawing with $A \cap B$ on the exterior face, it follows that $H[A - B]$ is connected. If $|A - B| \geq 2$, then the second conclusion follows trivially from local 5-connectivity. $\square$

For the splitter theorem, we need the following crucial but easy lemmas that roughly mean the following: "If a triangulation H has a surface minor isomorphic to a locally 5-connected triangulation G, and (A, B) is a local ≤ 3-separation of H, then $A - B$ contributes no vertex in the minor. If (A, B) is a ≤ 4-separation, then $A - B$ contributes at most one vertex in the minor."

Lemma 4.2.1 *Let G, H be triangulations of a surface Σ such that G is a surface minor of H. Let G be locally 5-connected, and let H have a local ≤ 3-separation (A, B). Then G is a surface minor of the triangulation obtained from H by deleting $A - B$.*

Proof: By Observation 9, H can be obtained from G by a sequence of splits. Thus vertices in G correspond to branch sets in H. It follows from the local 5-connectivity of G that no branch set is contained in $A - B$. Since G is a triangulation, the conclusion of the lemma easily follows. $\square$

Lemma 4.2.2 *Let G, H be triangulations of a surface Σ such that G is a surface minor of H. Let G be locally 5-connected, and let H have a local ≤ 4-separation (A, B). Then G is a surface minor of the graph obtained from H by replacing $A - B$ with a single vertex joined to every vertex in $A \cap B$.*

Proof: By Observation 9, H can be obtained from G by a sequence of splits. Thus vertices in G correspond to branch sets in H. It follows from the local 5-connectivity of G that $A - B$ contains at most one branch set. Since $H[A]$ has a planar drawing with $A \cap B$ on the infinite face, and since G is a triangulation, the conclusion of the lemma easily follows. $\square$

For an edge e, we define $\deg(e)$ to be the minimum of $\deg(x), \deg(y)$, where x and y are the end-points of e. Clearly, in a locally 5-connected triangulation, $\deg(e) \geq 4$ for every edge e.

The proof of the splitter theorem proceeds by considering two cases, depending on whether contracting an edge e in H creates a violating separation of order 3 or 4. The latter case uses the following technical lemma.

Lemma 4.2.3 *Let G and H be locally 5-connected triangulations of a surface Σ such that G is a surface minor of H. Let an edge e of H be such that H/e has a surface minor isomorphic to G. Let e be contained in a 5-cycle C that induces a local separation (A, B) with $|A - B| \geq 2$. Further, suppose that the 5-cycle C is chosen such that $|A - B|$ is maximum, and let H_0 be the triangulation obtained by replacing $A - B$ by a single vertex x joined to every vertex in $A \cap B$. Then one of the following outcomes must hold:*

1. *H_0 is locally 5-connected*

2. *there is an edge e' of H such that H/e' has a surface minor isomorphic to G, and* $\deg(e') < \deg(e)$

Proof: Let (A', B') be the local 4-separation in H/e, corresponding to the 5-separation (A, B) in H. By Lemma 4.2.2 applied to $G, H/e$, G is a surface minor of the triangulation obtained from H/e by replacing $A' - B'$ by a single vertex joined to every vertex of $A' \cap B'$. It is easy to see that the above triangulation is isomorphic to H_0/e. Thus G is a surface minor of H_0. In particular, H_0 has at least six vertices. We may assume that H_0 is not locally 5-connected, otherwise the outcome 1 of the lemma holds. Thus H_0 has a violating local $\leq$4-separation (A_0, B_0).

By the local 5-connectivity of H, $A_0 \cap B_0$ must contain x. Also, $A_0 - B_0$ and $B_0 - A_0$ must each contain at least one neighbor of x. (Those two neighbors, being on opposite sides of a separation, must be non-adjacent vertices of C.) Thus $A_0 \cap B_0$ must contain at least two of the neighbors of x. Now if x has exactly one neighbor v in $B_0 - A_0$, then $\big((A_0 - \{x\}) \cup (A - B) \cup \{v\}\,,\, B_0 - \{x\}\big)$ is a local separation that contradicts the local 5-connectivity of H. Thus it follows that x has exactly two neighbors v_3, v_4 in $B_0 - A_0$, exactly one neighbor v_1 in $A_0 - B_0$, and exactly two neighbors v_2, v_5 in $A_0 \cap B_0$ (such that $V(C)$ consists of $v_1, \ldots, v_5$, in that order). But then, consider the local separations (X_1, Y_1) and (X_2, Y_2) in H, where

$$X_1 = (A_0 - \{x\}) \cup (A - B) \cup \{v_3, v_4\} \qquad Y_1 = B_0 - \{x\}$$

$$X_2 = A_0 - \{x\} \qquad Y_2 = (B_0 - \{x\}) \cup (A - B) \cup \{v_1\}$$

Now $|X_1 \cap Y_1| = |A_0 \cap B_0| + 1$. Thus it follows that (X_1, Y_1) has order 5, (A_0, B_0) has order 4, and $|A_0 - B_0| \geq 2$ (since it violates local 5-connectivity, by assumption). But then (X_2, Y_2) is also a local 4-separation (in H), which means that $A_0 - B_0$ has exactly two vertices, where y is the vertex of $A_0 - B_0$ other than v_1. (The schematic in Figure 4.1 shows the separation (A_0, B_0) in H_0, where the dotted circle is a boundary curve for the separation. We follow the convention that solid vertices are not incident with any more edges than the ones indicated.) Applying Lemma 4.2.2 to G, H_0 and the separation (A_0, B_0), it follows that G is a surface minor of H_0/yv_1 (which in turn is a surface minor of H/yv_1).

Now e must be either v_1v_2 or v_1v_5. (If it is any of the other three edges in C, then the cycle $y, v_2, \ldots, v_5$ contradicts the choice of C.) By symmetry, we may assume that e

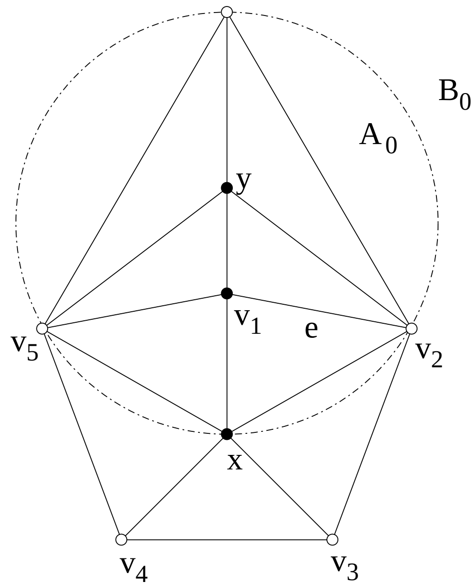

Figure 4.1: The local separation (A_0, B_0) in H_0

is the edge v_1v_2. Clearly, v_2 has degree ≥ 5 in H_0, and hence in H. In fact, v_1 also must have degree ≥ 5 in H, otherwise the separation (A_0, B_0) in H_0 will correspond to a local 4-separation in H that violates its local 5-connectivity.

Thus $\deg(e) \geq 5$ (in H). Outcome 2 of the lemma now follows with e' being the edge yv_1. □

4.3 The Main Theorem

Theorem 4.3.1 *Let G and H be locally 5-connected triangulations of a surface Σ such that G is a surface minor of H. Then H can be obtained from G by a sequence of vertex splits, such that every intermediate triangulation in the sequence is also locally 5-connected.*

Proof: By Observation 9, G can be obtained from H by a sequence of edge contractions. It suffices to show that the contractions can be chosen in such a way that every intermediate embedding stays locally 5-connected. We use induction on $|H| - |G|$ (that is, the number of contractions). For the base case, where $|H| - |G| = 1$, there is nothing to prove. Since $|G| \geq 6$ by local 5-connectivity, we may thus assume that $|H| \geq 8$.

Now there is an edge e in H such that G is a surface minor of H/e. Choose such an edge e with $\deg(e)$ minimum. If H/e is locally 5-connected, we are done by induction. Since H/e

has at least seven vertices, we may thus assume that H/e has a violating local separation (A', B'). Let (A, B) be the corresponding local separation in H.

First, suppose that (A, B) has order 5. Thus (A', B') has order 4, and $|A - B| = |A' - B'| \geq 2$. By Observation 10, it follows that e is in a 5-cycle C that induces (A, B). Choose C such that $|A - B|$ is maximum. Let H_0 be the triangulation obtained from H by replacing $A - B$ with a single vertex x joined to every vertex in C. By Lemma 4.2.2 applied to $G, H/e$, and the separation (A', B'), it follows that G is a surface minor of H_0/e. (In particular, G is a surface minor of H_0, and $|G| < |H_0|$.) Now apply Lemma 4.2.3 to G, H, and the edge e. If outcome 1 of the lemma holds, we can apply induction to G, H_0 and H_0, H. Thus we may assume that outcome 2 holds. But then the edge e' contradicts the choice of e.

Thus (A, B) has order 4 (and (A', B') has order 3). By the local 5-connectivity of H, it follows that $A - B$ has exactly one vertex u of degree 4, joined to every vertex in $A \cap B$. Label the vertices in $A \cap B$ as $v_1, \ldots, v_4$, in that cyclic order according to the rotation at u, such that $\deg(v_1) \leq \deg(v_i)$ for $i = 1, \ldots, 4$. Now, instead of contracting e in H, we contract the edge $e' = uv_1$. If H/e' is locally 5-connected, then we are done by induction. Also, since $\deg(e') = 4$, it follows from the argument in the previous paragraph that H/e' has no violating local 4-separation. Thus we may assume that it has a local 3-separation (A'', B'') with $A'' - B''$ non-empty. By the local 5-connectivity of H, it follows that $A'' - B''$ has a unique vertex, and that vertex must be one of v_2, v_4, say v_2. Let z be the fourth neighbor of v_2 (other than v_1, u, v_3). Figure 4.2 shows the neighborhoods of u and v_2 in H. (Again, we follow the convention that solid vertices are not incident with any more edges than the ones indicated.) Clearly, z is distinct from v_1, v_3 (but may be identical to v_4). If $z \neq v_4$, then the triangle zv_3v_4 yields a local 3-separation that contradicts the local 5-connectivity of H. Thus $z = v_4$. (In particular, the triangle v_2uv_4 traces a non-contractible curve). Notice that $\deg(v_2) = 4$ (in H), so by the numbering of $v_1, \ldots, v_4$, we conclude that $\deg(v_1) = 4$, and that the star of v_1 consists of the edges v_1u, v_1v_2, and two parallel edges incident with v_4. Moreover, the rotation at v_1 must be such that the two parallel v_1v_4 edges must be shared by some face, which contradicts the fact that H is a triangulation. This finishes the proof

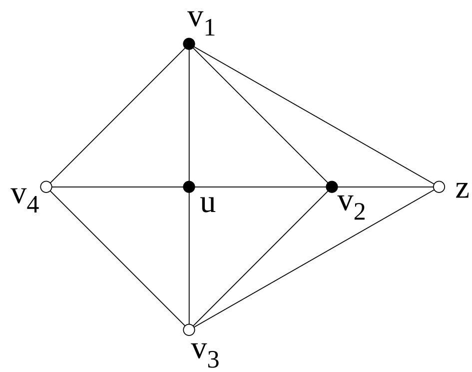

Figure 4.2: The neighborhoods of u and v_2 in H

of the theorem. □

From Lemma 4.1.1, we immediately obtain the following corollary:

Corollary 4.3.2 *Let G and H be simple locally 5-connected triangulations of a surface Σ such that G is a surface minor of H. Then H can be obtained from G by a sequence of strong splits. (Equivalently, every intermediate triangulation in the sequence is also locally 5-connected.)*

□

CHAPTER V

K_6 MINORS IN THE TORUS AND THE KLEIN BOTTLE

We use the notation from Chapter 4, except that graphs in this chapter are restricted to being simple, that is, they do not have parallel edges or loops.

5.1 K_6 *Minors in the Torus*

An edge in a (simple) triangulation is called *contractible* if it can be contracted (in the sense of a "surface contraction" defined in Chapter 4) without creating a loop or a parallel edge. Thus an edge is contractible if and only if it is not contained in any non-facial triangle.

A triangulation of the torus is called *irreducible* if it has no contractible edges. [31] gives a list of the irreducible triangulations of the torus. (This list was originally found by Duke and Grünbaum [12], but they did not publish their result.) There are twenty one triangulations in the list, up to isomorphism; they are shown in Figures 5.1, 5.2, and 5.3. (The figures use the usual unfolding of the torus into a rectangle. The torus is obtained by pasting opposite pairs of this rectangle.)

Theorem 5.1.1 ([12, 31]) *There are precisely 21 irreducible triangulations* $T^1, \ldots, T^{21}$ *of the torus, up to isomorphism.*

Theorem 5.1.2 *Every locally 5-connected (simple) triangulation of the torus, with the exception of* T^{21} *shown in Figure 5.3, has a minor isomorphic to* K_6.

Proof: Let H be a locally 5-connected triangulation of the torus not isomorphic to T^{21}. By Theorem 5.1.1, H has a surface minor isomorphic to one of the irreducible triangulations $T^1, \ldots, T^{21}$. Figures 5.1 and 5.2 show the triangulations $T^1, \ldots, T^{20}$. T^1 is the complete graph K_7, and the labeling on the remaining 19 triangulations demonstrates that they all have a K_6 minor. (The labels a through f represent the branch sets of the six vertices of

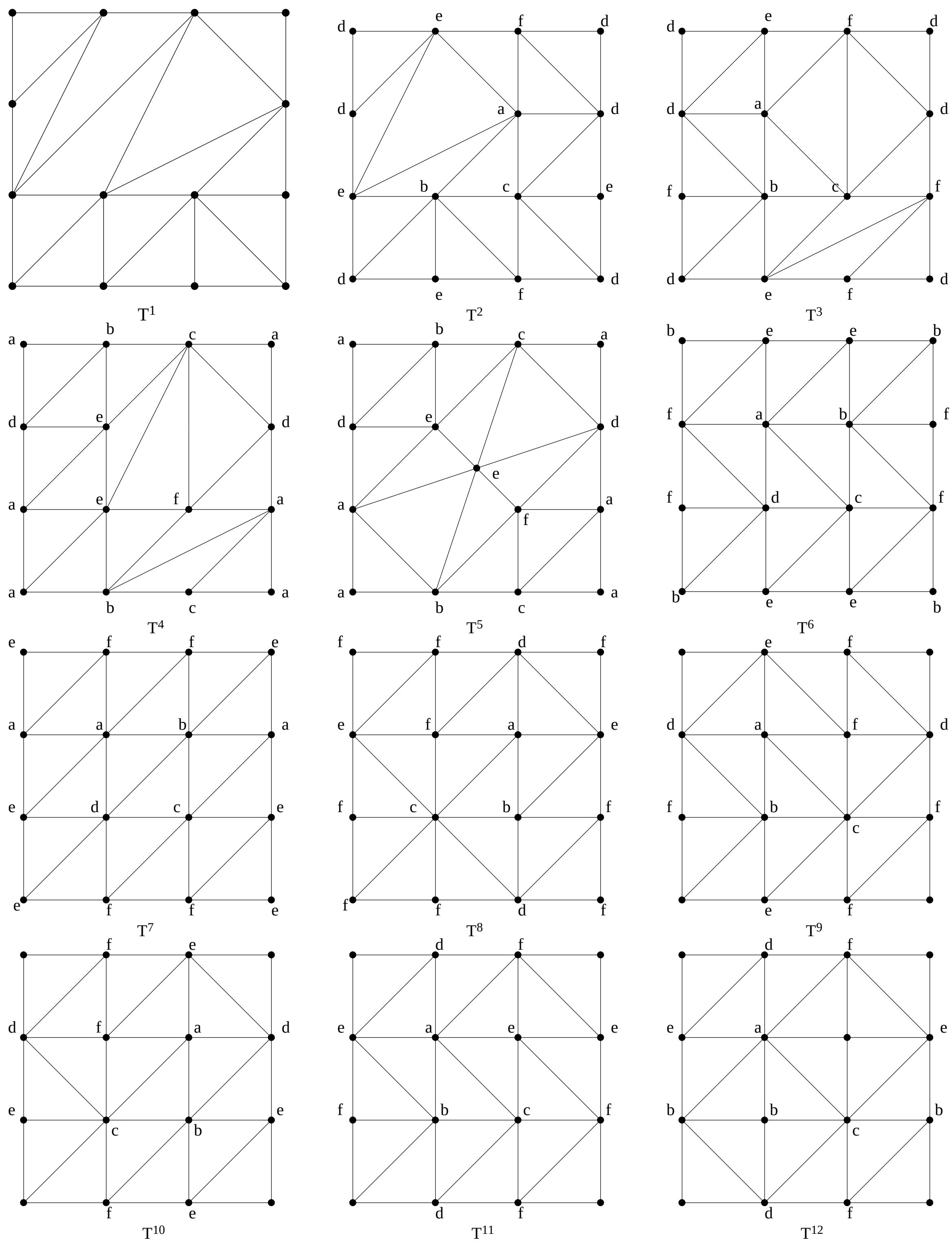

Figure 5.1: Triangulations $T^1, \ldots, T^{12}$ of the torus

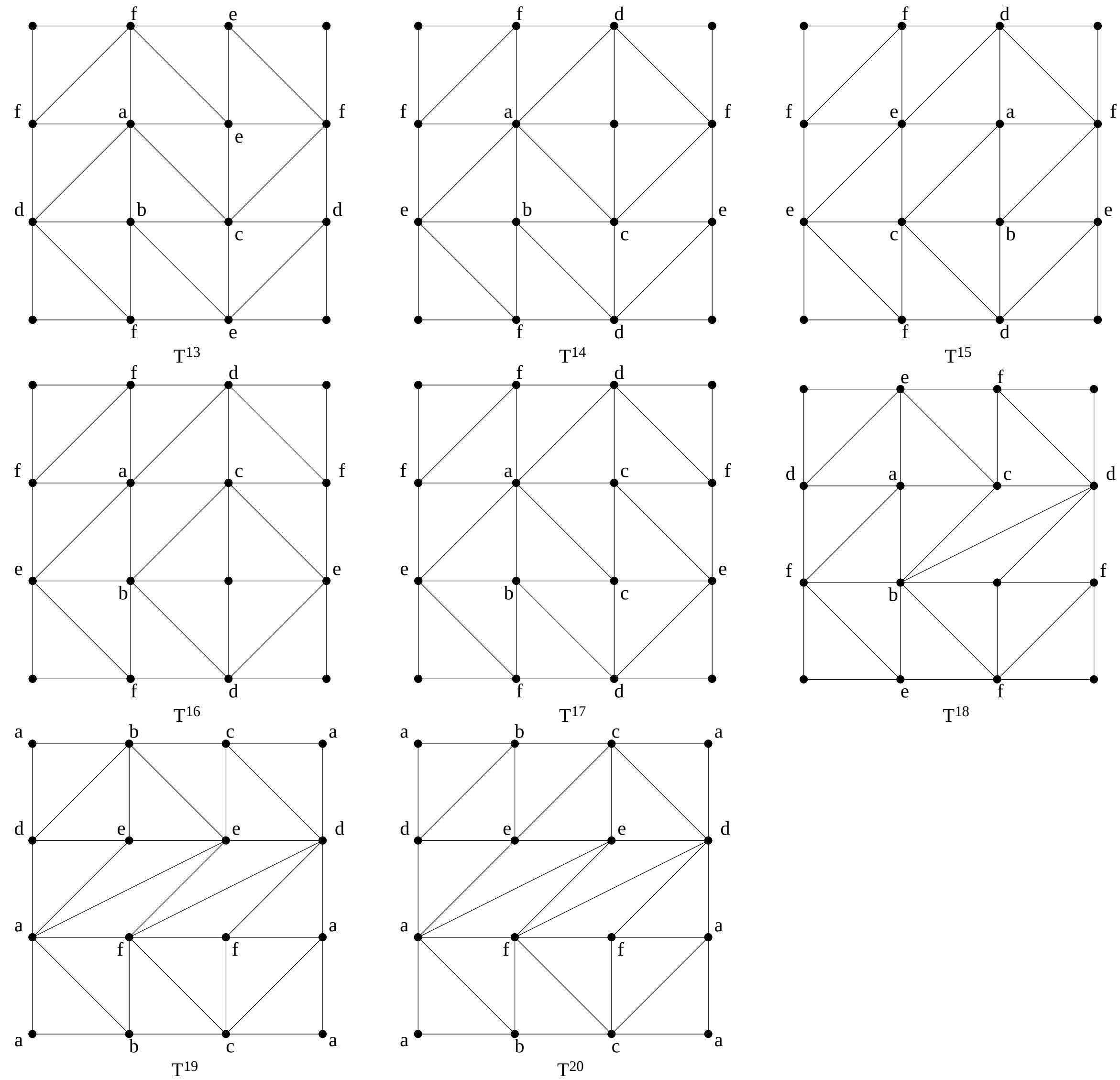

Figure 5.2: Triangulations $T^{13}, \ldots, T^{20}$ of the torus

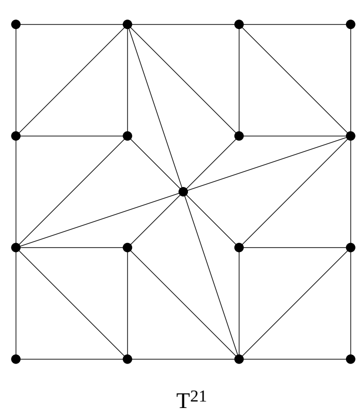

Figure 5.3: Triangulation T^{21} of the torus

K_6.) Thus if H has a surface minor isomorphic to any of these twenty triangulations, then H also has a K_6 minor.

Thus we may assume that H has a surface minor isomorphic to T^{21} (but is not itself isomorphic to T^{21}). By Corollary 4.3.2, it suffices to show that

1. T^{21} is locally 5-connected, and

2. Any triangulation obtained from T^{21} by applying a strong split has a K_6 minor

The vertices of T^{21} can be labeled $x_1, \ldots, x_5, y_1, \ldots, y_5$ such that for $i = 1, \ldots, 5$, x_i is adjacent to all other vertices except y_i. Let $X = \{x_1, \ldots, x_5\}$ and $Y = \{y_1, \ldots, y_5\}$.

T^{21} is 4-connected and simple, and hence has no local 3-separation (A, B) with $A - B$ non-empty. Suppose there is a local 4-separation (A, B) with $|A - B| \geq 2$. Clearly, $A - B$ is disjoint from X, and hence contains at least two vertices in Y. But then $A \cap B$ must contain X, a contradiction. Thus T^{21} is locally 5-connected.

The vertices in X are symmetric among themselves (and so are the vertices of Y). More precisely, for all i, j, there is an automorphism of T^{21} that maps x_i to x_j and respects the embedding (that is, maps faces to faces). Furthermore, the embedding is such that, for any x_i, its four neighbors in X and Y respectively appear alternately in the rotation at x_i.

We claim that, up to symmetry, there are exactly four ways to apply a strong split to T^{21}. By condition (i) of a strong split, none of the vertices in Y can be split, as they have degree 4 each. By symmetry, it suffices to consider any one vertex in X, say x_1. Let N_1 and N_2 be the two fans around x_1, as defined by the split, such that $|N_1| \leq |N_2|$. Now $|N_1| + |N_2| = \deg(x_1) + 2 = 10$, and by the condition (i) of a strong split, we have $|N_1|, |N_2| \geq 3$. Thus $|N_1|, |N_2|)$ is either $(3, 7)$, $(4, 6)$ or $(5, 5)$. It is easy to check that, up to symmetry, cases 1 and 2 lead to one type of split each, and case 3 leads to two types of splits.

Figure 5.4 shows, up to isomorphism, the four triangulations that can be obtained from T^{21} by applying a strong split, along with labellings that demonstrate a K_6 minor in each of them. This proves the theorem. □

A graph G is said to be *internally 5-connected* if (i) G is 4-connected and has at least

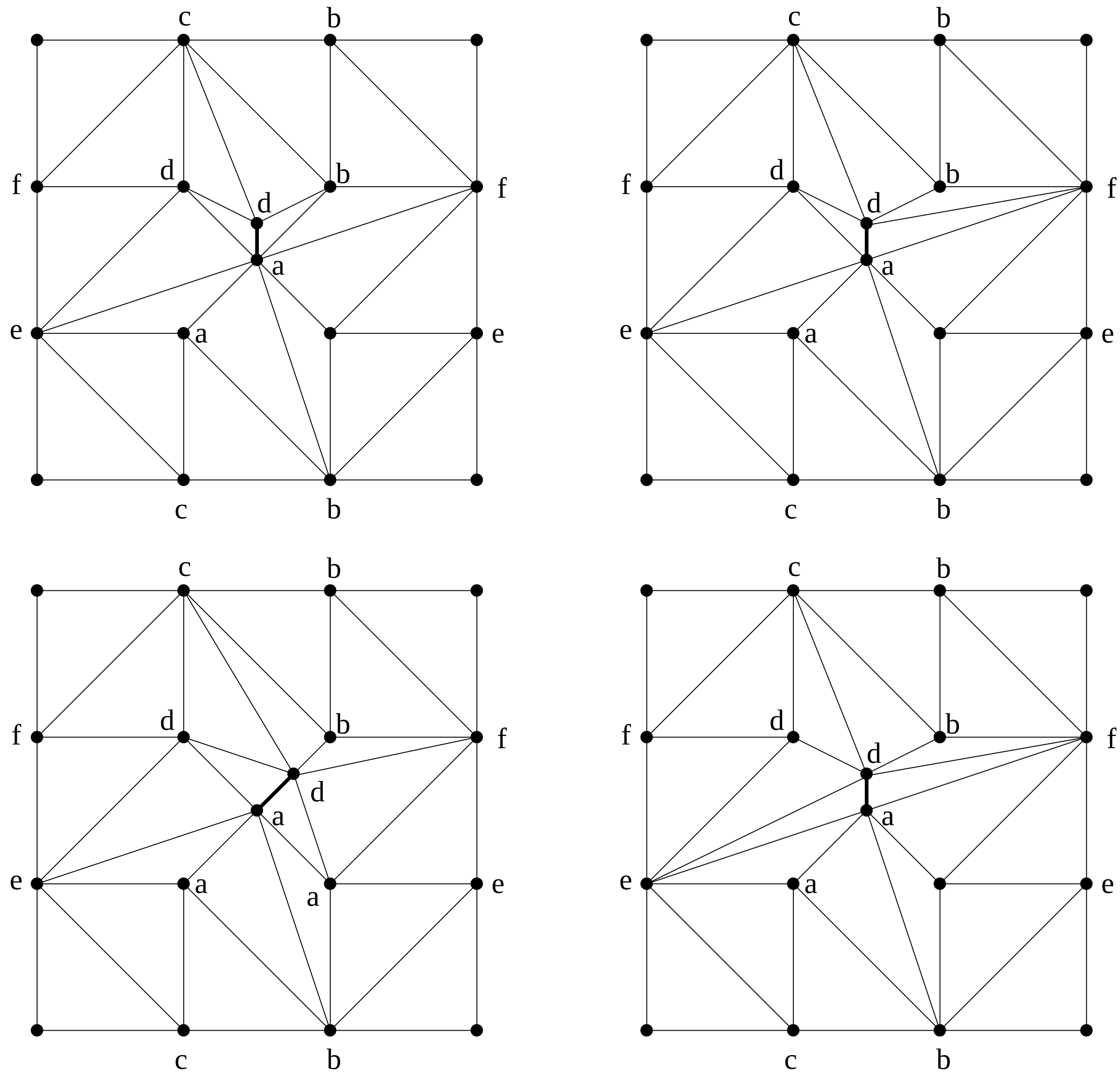

Figure 5.4: Strong splits in T^{21}, up to isomorphism

six vertices, and (ii) for every separation (A, B) of G of order at most four, at least one of $|A - B|, |B - A|$ is ≤ 1. For a simple graph embedding G, it is easy to check that internal 5-connectivity implies local 5-connectivity. Thus we have the following:

Corollary 5.1.3 *Every internally 5-connected (simple) triangulation of the torus, with the exception of T^{21} shown in Figure 5.3, has a minor isomorphic to K_6.*

□

5.2 K_6 *Minors in the Klein Bottle*

We now prove that every graph embedded in the Klein bottle with representativity at least 4 has a minor isomorphic to K_6.

The *representativity* of an embedding is the minimum k such that there is a non-contractible curve in the surface meeting the graph in at most k points. This invariant of the embedding is also called *face-width*. It is a measure of how locally planar the embedding is. (If representativity is $> k$, then any subgraph of diameter $< k$ must be planar.)

A simple closed curve C in a surface Σ is called *separating* if $\Sigma \setminus C$ consists of at least two regions, and is called *non-separating* otherwise.

A simple closed curve is called *orientation-preserving* if traversing it once preserves the sense of orientation (that is, clockwise or anti-clockwise); it is called *orientation-reversing* otherwise.

The Klein bottle can be obtained from identifying the pairs of opposite sides of a rectangle as shown in Figure 5.5. There are three kinds of non-contractible simple closed curves

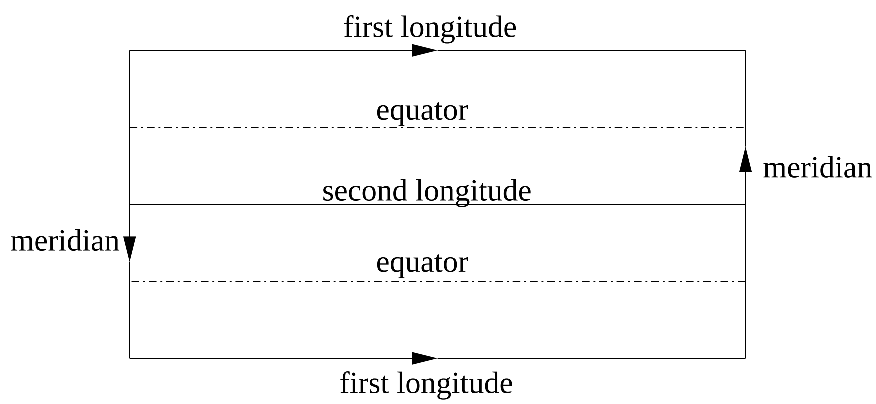

Figure 5.5: Non-contractible simple closed curves in the Klein bottle

in the Klein bottle, up to homeomorphism:

1. Orientation-preserving, non-separating curves called *meridians*: Any two meridians are homotopic to each other.

2. Orientation-preserving, separating curves called *equators*: Any two equators are also homotopic to each other. (The union of the two dotted arcs in Figure 5.5 forms one such equator.)

3. Orientation-reversing, non-separating curves called *longitudes*: Any two longitudes are homeomorphic, but not necessarily homotopic to each other. Every maximal family of disjoint longitudes has size two. (The curves "first longitude" and "second longitude" in Figure 5.5 comprise one such family. Any other longitude in the Klein bottle is homotopic to exactly one of them.) Furthermore, if a simple closed curve crosses a meridian exactly once, then it must be a longitude.

The proof of Theorem 5.2.3 relies on reducing the problem to a graph in the projective plane. To this end, we need the following lemma, a variant of which was shown in [46].

Given a graph G embedded in the Klein bottle, a cycle in G is called *equatorial* (respectively, *longitudinal*) if the simple closed curve that it traces is an equator (respectively, a longitude) in the Klein bottle.

Lemma 5.2.1 *Let G be graph embedded in the Klein bottle with representativity at least 4. Then G satisfies at least one of the following outcomes:*

1. *G has two vertex-disjoint equatorial cycles E_1, E_2*

2. *G has two longitudinal cycles L_1, L_2 and an equatorial cycle E such that the three cycles are pairwise vertex-disjoint*

Proof: Choose a meridian M in the Klein bottle such that M meets G in as few points as possible. Without loss of generality, we may assume that the points of intersection are all vertices of G. Let $v_1, \ldots, v_k$ be those vertices. By hypothesis, $k \geq 4$. Cutting open the Klein bottle along M yields an embedding G' in a cylinder, such that the vertices on the two boundary components are, respectively, $x_1, \ldots, x_k$ and $y_k, \ldots, y_1$ (where the vertices appear in those cyclic orders along the respective components, for some fixed orientation around the axis of the cylinder). G can be obtained from G' by identifying x_i, y_i into the vertex v_i, for $i = 1, \ldots, k$.

We claim that in G' there are k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i has ends x_i and $y_{\sigma(i)}$, for $i = 1, \ldots, k$. Indeed, by Menger's theorem, if these paths do not exist, then there exists a meridian M' in the Klein bottle that meets G in fewer than k points, contrary to the choice of M. This proves the claim.

Since the paths P_i are pairwise disjoint, it follows that $\sigma(1), \ldots, \sigma(k)$ is simply a cyclic shift of $k, \ldots, 1$. It is easy to check that $\sigma(\sigma(i)) = i$, for $i = 1, \ldots, k$. If $\sigma(i) = i$, then the path $P_i = P_{\sigma(i)}$, in G, forms a cycle that crosses M exactly once, and is hence longitudinal. On the other hand, if $\sigma(i) \neq i$, then $P_i \cup P_{\sigma(i)}$, in G, forms a cycle that crosses M twice, and furthermore traces a separating curve in the Klein bottle. Thus that cycle must be equatorial. Since $k \geq 4$, the conclusion of the lemma now follows easily. $\square$

We now reduce the problem on the Klein bottle to one on the projective plane:

Lemma 5.2.2 *Let G be a graph embedded in the Klein bottle with representativity at least 4. Then G has a minor isomorphic to a graph G' embedded in the projective plane with representativity at least 4.*

Proof: By Lemma 5.2.1, G satisfies one of the outcomes in that lemma.

First, suppose G satisfies outcome 1, and has equatorial cycles E_1, E_2 as specified. The curve traced by E_1 separates the Klein bottle into two regions. Let R_1, R_2 be the closures of those two regions. (Both R_1 and R_2 are homeomorphic to a Möbius band.) Since E_2 is vertex-disjoint from E_1, $V(E_2)$ is contained in $V(G) \cap R_i$ for some $i = 1, 2$ (say, $i = 1$). Let G' be the graph determined by the vertices and edges contained in R_1. Indeed, by attaching a disk to R_1 along E_1, we get an embedding G' in the projective plane (with E_1 being one of the facial cycles). We claim that G' has representativity at least 4. Suppose not, and let C be a non-contractible simple closed curve in the projective plane that meeting G' in at most 3 points. It follows that C must meet the interior of the face bounded by E_1 (otherwise, the corresponding curve in the Klein bottle violates the representativity of G.) But then C would have to meet each of E_1, E_2 in two distinct points, which is a contradiction. This proves our claim.

Now suppose that G satisfies outcome 2 of Lemma 5.2.1 and has cycles L_1, L_2 and E as specified. Contract the cycle L_2 to a single vertex v. It is an elementary topological argument to show that the resulting graph has an embedding in the projective plane. Let G' be the embedding so obtained. We claim that G' has representativity at least 4. Suppose not, and let C be a non-contractible simple closed curve in the projective plane that meets

G' in at most 3 points. Since G has representativity 4, C must contain the vertex v. Also, it meets L_1 in exactly one vertex. (Every two non-contractible curves in the projective plane cross at exactly one point.) In doing so, C would have to meet E in at least two points. Thus C meets G' in at least 4 points, a contradiction. This proves the lemma. □

We are now ready to prove the theorem:

Theorem 5.2.3 *Every graph G embedded in the Klein bottle with representativity at least 4 has a minor isomorphic to K_6.*

Proof: By Lemma 5.2.2, G has a minor isomorphic to a graph G' embedded in the projective plane with representativity at least 4. By Lemma 5.2.5, G' has a minor isomorphic to K_6, and the theorem follows. □

To prove Lemma 5.2.5, we need the following two well-known operations on graph embeddings: (i) a $Y\Delta$-*operation* deletes a vertex of degree 3, and adds a (facial) triangle passing through its 3 neighbors, while (ii) a ΔY-operation deletes a facial triangle and adds a new vertex joined to the 3 vertices.

Randby (see [47]) proved the following theorem:

Theorem 5.2.4 *Every graph embedded in the projective plane with representativity at least 4 has a minor isomorphic to one of the graphs obtained from the 4×4-grid in the projective plane, shown in Figure 5.6, by a sequence of $Y\Delta$ and ΔY-operations.*

Lemma 5.2.5 *Every graph G' embedded in the projective plane with representativity at least 4 has a minor isomorphic to K_6.*

Proof: By Theorem 5.2.4, G' has a minor isomorphic to one of the graphs obtained from the 4×4-grid by a sequence of $Y\Delta$ and ΔY-operations. There are 270 such graphs, and each of them has a K_6 minor. The generation of the list, and the verification of a K_6 minor in each graph therein, was done with the help of computer programs. It now follows that G' too must have a K_6 minor. □

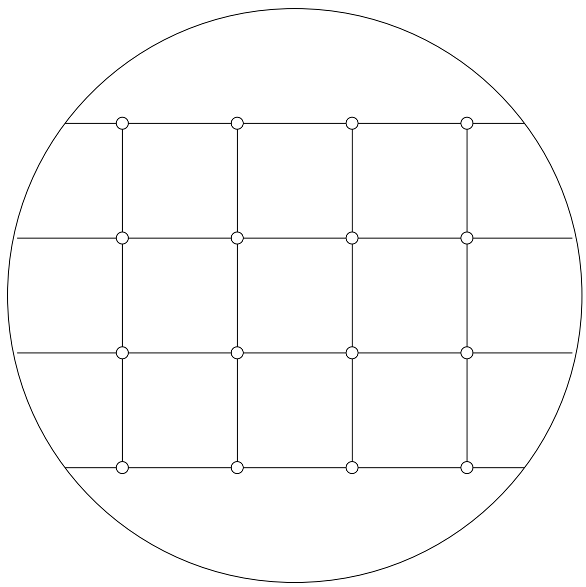

Figure 5.6: The 4×4-grid in the projective plane

CHAPTER VI

FINDING 3-SHREDDERS EFFICIENTLY

6.1 Introduction

Connectivity is an important invariant of graphs. Efficient algorithms for determining the connectivity properties of a graph are both theoretically interesting and practically useful. For general k, the fastest algorithm to test for k-vertex connectivity runs in time $O((n + \min(k^{5/2}, kn^{3/4}))m)$ [16]. The problem of counting the number of k-vertex cuts in a k-connected graph, for general k, however, is #P-complete [8].

For $k \leq 3$, linear time algorithms are known for testing k-vertex-connectivity and for finding the vertex cuts of size $k-1$. The cases $k = 1, 2$ are easily solved using depth-first search. For $k = 3$, see [22] and [20]. (The classical 3-connectivity algorithm of Hopcroft and Tarjan [22] finds, in $O(n + m)$ time, a decomposition into 3-connected components. One could then read off all the 2-cuts in the graph by looking at the decomposition.) For $k = 4$, [25] gives an almost-linear time algorithm for testing 4-connectivity and maintaining a decomposition into 4-connected components online, under certain update operations.

In this chapter, we consider those vertex cuts that not only disconnect the graph, but do so into at least three components. More precisely, a *shredder* in a graph G is a vertex cut S such that $G \setminus S$ has at least three components. A shredder having k vertices is called a *k-shredder*. For instance, if G is a tree, the 1-shredders of G correspond precisely to the vertices of degree at least three. One application of shredders is in node connectivity augmentation; (see [8]), for instance. [8] also presents an algorithm to find, for general k, the set of k-shredders of a k-vertex connected graph on n vertices that runs in time $O(k^2n^2 + k^3n^{1.5})$.

This chapter presents an algorithm to find the set of 3-*shredders* of a 3-*vertex connected graph* in time proportional to the number of vertices and edges in the graph. (The connectivity assumption is without loss of generality, because of the tri-connectivity algorithm in [22].) The best known current bound for this problem is $O(n^2)$, which follows from the general k-shredders algorithm in [8].

For a discussion of the motivation of this problem in the context of Graph Structure Theory, refer to Section 1.2.5.

6.2 Notation

Given a simple undirected graph G with n vertices and m edges, we can test for 3-connectivity in linear ($O(n+m)$) time using the algorithm in [22] (see also [20].) The 3-shredders algorithm will then proceed in several steps. In the first step, we generate a certain set of triples of vertices that includes all the 3-shredders. In subsequent steps, we eliminate those triples that are not 3-shredders. The basic strategy for these steps is depth-first search (dfs). We first find a depth-first spanning tree T, starting at an arbitrary vertex, which will henceforth be called the root. The edges of T (tree edges) will be directed from parent to child, and the remaining edges (back edges) will be directed from descendant to ancestor. We denote tree edges by $u \to v$ and back edges by $u \hookrightarrow v$. The adjacency list $\mathrm{Adj}(u)$ denotes the set of all edges with tail u. $\mathrm{Adj}^R(u)$ will denote the set of *back edges* with head u. (We consider the tree T to be "growing downwards", with the root being the top vertex, and the children of a vertex u being listed from left to right according to their order in $\mathrm{Adj}(u)$.) The generation procedure and the subsequent steps will have the general format given in Table 6.1.

It is important that during the generation procedure and the subsequent steps, the edges in an adjacency list are processed in a specific order. Before we describe what the order is, we need to define the quantities HIGH1, HIGH2 and HIGH3, which are defined for all edges of G, and for all vertices except for the root of T. The value of HIGH1, HIGH2 or HIGH3 is either a vertex of G, or as a special case, infinity. (Vertices are later going to be identified with their post-order numbers with respect to a certain depth-first traversal of G; the ∞ notation is intended to be consistent with that numbering.) It should be noted that the concepts of HIGH1 and HIGH2 are more or less the same as "LOWPT1" and "LOWPT2", introduced in [22], except when they are defined to be infinity.

Table 6.1: General Format for the Dfs-based Steps

```
A: (statement to be inserted here)
dfs_step(root);

procedure dfs_step(u)
begin
    for e ∈ Adj(u) do begin
        forward_visit(e)
        if e is a tree edge u → v then begin
            dfs_step(v)
            backward_visit(e)
        end
    end
    B: (statement to be inserted here)
end
```

For $v \in V$, let $D(v)$ be the set of descendants of v in the depth-first spanning tree (including v itself), and let $ND(v) := |D(v)|$. (With a slight abuse of notation, we sometimes refer to $D(v)$ as a "subtree" rather than the vertex set of a subtree.) If e is a back edge $u \hookrightarrow v$, we define $\text{HIGH1}(e) = v$ and $\text{HIGH2}(e) = \text{HIGH3}(e) = u$. Now let e be a tree edge $u \rightarrow v$. We call a vertex a an *attachment* of the subtree $D(v)$ if it is a proper ancestor of u and $v' \hookrightarrow a$ for some $v' \in D(v)$. HIGH1(e) is defined as the attachment of $D(v)$ that is highest in the tree, i.e. closest to the root. If no such attachment exists, we define $\text{HIGH1}(e) = \infty$. HIGH2(e) is defined as the second highest attachment of $D(v)$ (∞ if no such attachment exists). Finally, HIGH3(e) is defined as the third highest attachment of $D(v)$ (∞ if no such attachment exists.) Note that since G is 3-connected, $\text{HIGH1}(u \rightarrow v) \neq \infty$ unless u is the root and v is its (unique) child. Similarly, $\text{HIGH2}(u \rightarrow v) \neq \infty$ unless u is the root or its child, and v is its (unique) child.

For a vertex v that is not the root, we denote by HIGH1(v) the value of HIGH1 for the (unique) tree edge $u \rightarrow v$. The quantities HIGH1, HIGH2 and HIGH3 can be easily computed in a bottom-up fashion by a dfs.

We are now ready to describe the order on Adj(u). An edge e will precede an edge f in Adj(u) if either HIGH1(e) is higher in the tree than HIGH1(f), or $\text{HIGH1}(e) = \text{HIGH1}(f)$

and HIGH2(e) is higher than HIGH2(f), or HIGH1(e) = HIGH1(f) and HIGH2(e) = HIGH2(f) and HIGH3(e) is higher than HIGH3(f). (∞ is considered higher in the tree than any vertex. Ties are broken arbitrarily in the above order.) The adjacency lists can be sorted in $O(n+m)$ time using a slight modification of radix sort with $n+1$ buckets, and future depth-first searches will use this ordering to process the edges in an adjacency list. The ordering of the adjacency lists here is somewhat similar to that described in [22], but differs in two respects. Firstly, the ordering here is lexicographic with respect to three quantities instead of two, as is the case in [22]. Secondly, the ∞ part of the definition of HIGH1, HIGH2 etc. ensures the following: Among all edges of Adj(u) having some fixed value of HIGH1, back edges will appear in the front of the ordering. Similarly, among all tree edges in Adj(u) having some fixed values for HIGH1 and HIGH2, tree edges with HIGH3 being ∞ (that is, those for which the corresponding subtree has only two attachments) appear in the front of the ordering. This fact turns out to be useful for the 3-shredders algorithm, whereas it is immaterial in the case of the 3-connectivity algorithm in [22]. In fact, with respect to the ordering in [22], the above-mentioned edges appear at the end of the respective sublists of Adj(u).

The vertices are then numbered 1 through n in the order in which they are *last* examined by a dfs (using the new ordering on the adjacency lists.) Henceforth, we will identify the vertices with their post-order number as given above, and refer to a vertex and its integer label interchangeably. For instance, a range of integers can be construed as a subset of $V(G)$ (if it falls between 1 and n.) The quantities HIGH1, HIGH2 and HIGH3 may also be treated as integers with respect to the above numbering. Note that the numbering *respects height on the tree* i.e. if u is an ancestor of v ($u \rightarrow^* v$), then $u \geq v$. (We use "$\rightarrow^*$" to denote a path of 0 or more tree edges.)

The first edge in an adjacency list is called a *leftmost edge*. We call v a *leftmost vertex* if it is not the root and the (unique) tree edge $u \rightarrow v$ is leftmost. Otherwise, we call v non-leftmost. A path consisting of leftmost edges is called a leftmost path. If $u \rightarrow^* v$ is a leftmost path, v is called a leftmost descendant of u.

We need to define two more quantities, LOW1 and RCH (for "reach"), as follows. Let

$e = (u \to v)$ be a tree edge. We define LOW1(e) to be the *lowest* attachment of $D(v)$ distinct from u. (By lowest, we mean farthest from the root.) If no such attachment exists, define LOW1 to be 0. Note that if the graph is 2-connected, every edge $u \to v$ has a non-zero LOW1 value unless u is the root and v its (unique) child. Section 6.4.1 describes how to compute LOW1 for all tree edges using a Union-Find procedure.

Further, we define, for every vertex u, RCH$(u) = \min\{w \mid w \hookrightarrow u\}$, where the minimum is ∞ if the set is empty. It is easy to compute RCH(u) for all vertices u in a bottom-up fashion in a dfs.

Next we need a few observations about the possible arrangement of the vertices of a 3-shredder in G, with respect to the dfs tree. We say that two vertices are comparable under the ancestor relation if one of them is an ancestor of the other (in the dfs tree.) It can be seen that if three vertices a_1, a_2 and a_3 are not mutually comparable under the ancestor relation i.e. they are not all on a (directed) path of tree edges, then $G \backslash \{a_1, a_2, a_3\}$ has at most two components. In other words, we have the following:

Lemma 6.2.1 *When G is 3-connected, a* 3*-shredder in G is always of the form* (a_1, a_2, a_3) *with $a_1 \to^* a_2 \to^* a_3$.*

Proof: Suppose not. Note that, since G is 3-connected, $\{a_1, a_2, a_3\}$ is a minimal vertex-cut, hence each of the three vertices is adjacent to some vertex from each of the components of $G \backslash \{a_1, a_2, a_3\}$. Now we may assume, without loss of generality, that one of the following three situations occurs:

No two of the three vertices are comparable. In this case, define V_0 to be $V(G) \setminus (\cup_{i=1}^{3} D(a_i))$ and V_i to be $D(a_i) \backslash \{a_i\}$, for $i = 1, \ldots, 3$. Clearly, V_0 is non-empty, and spans a connected subgraph of $G \backslash \{a_1, a_2, a_3\}$. For $i = 1, 2, 3$, V_i consists of the vertex sets of subtrees rooted at the children of a_i. Since a_i is not a cut-vertex, each of these subtrees has an attachment in V_0. But then, this means that $G \backslash \{a_1, a_2, a_3\}$ is connected, which is a contradiction.

Now suppose $a_1 \to^* a_2$ but a_3 is not comparable to a_1 or a_2. Since a_3 is not a cut-vertex, for every child w of a_3, $D(w)$ has an attachment that is a proper ancestor of a_3. Since every neighbor of a_3 is either a descendant or an ancestor of a_3, it follows that a_3 is adjacent to only one component of $G\backslash\{a_1, a_2, a_3\}$, which is a contradiction.

Finally, suppose a_1 is an ancestor of both a_2 and a_3, but a_2 and a_3 are mutually incomparable. Let a_1^2 be the child of a_1 that is an ancestor of a_2. Define V_0 to be $V(G) \setminus D(a_1)$ and V_1^2 to be $D(a_1^2) \setminus D(a_2)$. Since $\{a_1, a_2\}$ is not a vertex-cut, for every child w of a_2, $D(w)$ has an attachment in either V_0 or V_1^2. It follows that a_2 is adjacent to at most two components of $G\backslash\{a_1, a_2, a_3\}$, which is a contradiction. □

Henceforth, we shall consider a potential 3-shredder as an ordered triple of vertices, where the ordering refers to that along the tree path. Now let (a, b, c) be *any* triple of distinct vertices with $a \to p \to^* b \to q \to^* c$. With respect to the above triple (a, b, c), we define the following sets of vertices (refer to Figure 6.1):

- $\mathcal{A} = A \cup A' \cup A''$, where

  $A = V(G) \setminus (D(p) \cup \{a\})$

  A' consists of all subtrees of the form $D(v)$ where v is a child of b different from q and with HIGH1$(v) > a$

  A'' consists of all subtrees of the form $D(v)$ where v is a child of c with HIGH1$(v) > a$

- $\mathcal{B} = B \cup B' \cup B''$, where

  B consists of $D(p) \setminus D(b)$

  B' consists of all subtrees of the form $D(v)$ where v is a child of b different from q with HIGH1$(v) \leq a$

  B'' consists of all subtrees of the form $D(v)$ where v is a child of c with ($b <$ HIGH1$(v) < a$) OR (HIGH1$(v) = a$ AND $b <$ HIGH2$(v) < a$)

- $\mathcal{C} = C \cup C'$, where

  C consists of $D(q) \setminus D(c)$

  C' consists of all subtrees of the form $D(v)$ where v is a child of c with (HIGH1$(v) \leq b$) OR (HIGH1$(v) = a$ AND $c <$ HIGH2$(v) < b$) OR (HIGH1$(v) = a$ AND HIGH2$(v) = b$ AND $c <$ HIGH3$(v) < b$)

- $\mathcal{D}$, which consists of subtrees of the form $D(v)$ where v is a child of c with HIGH1$(v) =$

a, HIGH2$(v) = b$ and HIGH3$(v) = \infty$. These subtrees, if any, are clearly components of $G \backslash \{a, b, c\}$ by themselves, and we will refer to these as the *singular* components of the triple (a, b, c)

Lemma 6.2.2 *The sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, and the vertex sets of the singular components (defined above) all span connected subgraphs of $G \backslash \{a, b, c\}$. Further, these sets partition $V(G) \setminus \{a, b, c\}$.*

Proof: Denote $G \backslash \{a, b, c\}$ by G'. Since a is not a cut-vertex, A spans a connected subgraph of G'. It now follows by definition that $\mathcal{A}$ also spans a connected subgraph of G'.
Clearly, B spans a connected subgraph of G'. It follows by definition, that B' (respectively, B'') consists of the vertex sets of those subtrees rooted at children of b (respectively, c) that have an attachment in B but are not included in A' (respectively, A''). It follows that $\mathcal{B}$ spans a connected subgraph of G', and is disjoint from $\mathcal{A}$. A similar argument shows that $\mathcal{C}$ spans a connected subgraph of G', and is disjoint from $\mathcal{A}$ and $\mathcal{B}$. The lemma now follows. $\square$

A triple as above (respectively, a shredder) that has a singular component is called a *singular* triple (or shredder). Conversely, a triple (shredder) that does not have any singular components is called *non-singular*. Further, a singular triple (shredder) is called *degenerate* if HIGH1$(q) \leq a$, and *non-degenerate* otherwise. Note that for a non-singular triple (a, b, c), the maximum number of components of $G \backslash \{a, b, c\}$ is three (which is achieved only when $\mathcal{A}$,$\mathcal{B}$ and $\mathcal{C}$ all span distinct components of $G \backslash \{a, b, c\}$.) Hence, if the triple is to be a 3-shredder, then there must not be any edge between the vertex sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. By "X-Y edge", for $X, Y \subseteq V(G)$, we mean an edge with one end in X and the other end in Y, disregarding the direction that we are associating with the edges.

The above decomposition of $V(G) \setminus \{a, b, c\}$ divides the set of proper descendants of c into A'', B'', C' and $\mathcal{D}$. The ordering of Adj(c) (in particular, the ordering of c's children in the list) implies that the subtrees in A'' occur before all the other subtrees. We define the "corner" vertex α for the triple (a, b, c) as the "lower left corner" of the first subtree not in A'', i.e. α is the lowest numbered vertex, among the descendants of c, that is not in A''.

More precisely, if there exists a child v of c with $\mathrm{HIGH1}(v) \leq a$, let v_0 be the first[1] such v and $\alpha = v_0 - ND(v_0) + 1$, otherwise $\alpha = c$. It follows that $A'' = [c - ND(c) + 1, \alpha)$.

We define corner(e) for an edge $e \in \mathrm{Adj}(u)$ as follows: let $e' = u \rightarrow v'$ be the first *tree* edge to follow e in $\mathrm{Adj}(u)$ (if e is itself a tree edge, $e' = e$). We set corner$(e) = v' - ND(v') + 1$ (If no such v' exists, we set corner(e) to u.) The idea here is that, in the generation step, when we explore an edge $e \in \mathrm{Adj}(u)$, we generate part of a triple with $c = u$ and $a = \mathrm{HIGH1}(e)$. The value of corner(e), as defined above, gives us the right value of α for

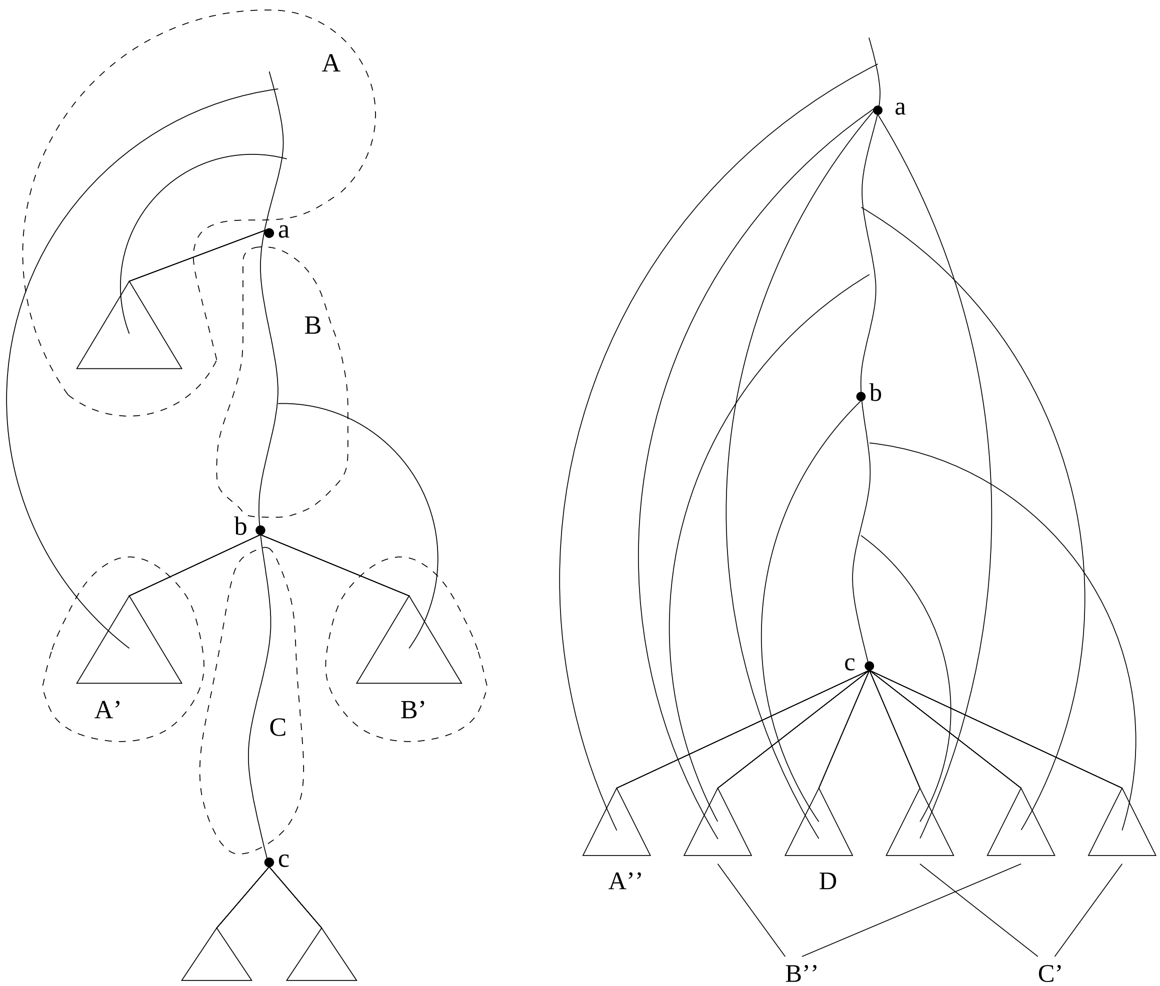

Figure 6.1: Potential Components of $G \backslash \{a, b, c\}$ (triangles denote subtrees)

this triple. (Note that the definition of α for a triple (a, b, c) does not involve b.)

We will use the following basic lemmas about any triple (a, b, c) of vertices with $a \rightarrow$

[1] "first" refers to the usual ordering of the adjacency lists

$p \to^* b \to q \to^* c$.

Lemma 6.2.3 *If the path $p \to^* b$ is not leftmost, then there is an A-B edge. In particular, for a non-singular shredder (a, b, c), the path $p \to^* b$ is leftmost.*

Proof: Suppose the path $p \to^* b$ is not leftmost, and let $s \to t$ be the first non-leftmost edge in it. Now $\text{HIGH1}(p) > a$ since a is not a cut-vertex, and since s is a leftmost descendant of p, $\text{HIGH1}(s) > a$. (In general, if y is a leftmost descendant of x, it is easy to see that $\text{HIGH1}(y) = \text{HIGH1}(x)$, assuming $\text{HIGH1}(x)$ is defined i.e x is not the root.) Let e be the first edge in $\text{Adj}(s)$. It follows that $\text{HIGH1}(e) > a$, which means there is an A-B edge. The latter inference about non-singular shredders follows from the fact that for a non-singular triple (a, b, c) to be a 3-shredder, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ must be vertex sets of distinct components of $G \backslash \{a, b, c\}$. □

Lemma 6.2.4 *Either $\text{HIGH1}(c) > a$, or there is an $\mathcal{A}$-$\mathcal{B}$ or $\mathcal{A}$-$\mathcal{C}$ edge. In particular, for a non-singular shredder (a, b, c), $\text{HIGH1}(c) > a$.*

Proof: If $\text{HIGH1}(c) \leq a$, then c cannot be adjacent to any vertex in $\mathcal{A}$, and hence $\mathcal{A}$ is not the vertex set of a component of $G \backslash \{a, b, c\}$ by itself. It follows that there must be an $\mathcal{A}$-$\mathcal{B}$ or $\mathcal{A}$-$\mathcal{C}$ edge. □

Lemma 6.2.5 *If the path $q \to^* c$ is not leftmost, then either $\text{HIGH1}(q) \leq a$ or there is an A-C edge. In particular, for a non-singular shredder (a, b, c), the path $q \to^* c$ is leftmost.*

Proof: The proof of the first statement is similar to the proof of Lemma 6.2.3. For the latter inference, note that for a non-singular shredder (a, b, c), it follows, from Lemma 6.2.4, that $\text{HIGH1}(c)$ (and hence $\text{HIGH1}(q)$) must be greater than a, and hence the path $q \to^* c$ is leftmost. □

For the following lemmas, let (a, b, c) be a non-singular shredder.

Lemma 6.2.6 *B'' is non-empty or there is a back edge $c \hookrightarrow v$ with $b < v < a$.*

Proof: This follows from the fact that c must be adjacent to a vertex in $\mathcal{B}$. □

Lemma 6.2.7 *One (or both) of the following conditions must hold:*

(i) $a = \mathrm{HIGH1}(e)$ *for some edge* e *in Adj(c)*

(ii) $\exists u$ *on the path* $q \rightarrow^* c$, $u \neq c$, *s.t.* $a = \mathrm{HIGH1}(e)$ *for some non-leftmost edge* e *in Adj(u).*

Proof: This follows from the fact that a must be adjacent to a vertex in $\mathcal{C}$. □

Lemma 6.2.8 *One (or both) of the following conditions must hold:*

(i) $\exists\ e \neq (b \rightarrow q)$ *in Adj(b) s.t.* $\mathrm{HIGH1}(e) > a$. *(In particular, if* $b \rightarrow q$ *is not leftmost, then this condition is automatically satisfied.)*

(ii) $\exists v \in A''$ *with* $v \hookrightarrow b$ *(i.e.* b *"sees" a back edge from a vertex in* A''*). In particular, if* $b \rightarrow q$ *is a leftmost edge, i.e. if* $b \rightarrow^* c$ *is a leftmost path, then this condition is equivalent to saying* $\mathrm{RCH}(b) < \alpha$.

Proof: This follows from the fact that b must be adjacent to a vertex in $\mathcal{A}$. □

Finally, we need the following lemma about degenerate shredders.

Lemma 6.2.9 *If* (a, b, c) *is a (singular and) degenerate shredder, then it has no* $\mathcal{A}$-$\mathcal{C}$ *edges, and it must have an* $\mathcal{A}$-$\mathcal{B}$ *edge.*

Proof: By definition, a degenerate shredder has $HIGH1(c) \leq HIGH1(q) \leq a$. It is easy to see then that there cannot be any $\mathcal{A}$-$\mathcal{C}$ edges, and that c cannot be adjacent to any vertex in $\mathcal{A}$. It follows that $\mathcal{A}$ cannot be the vertex set of a component of $G \setminus \{a, b, c\}$ by itself, and hence there must be an $\mathcal{A}$-$\mathcal{B}$ edge. □

6.3 The Generation Step

As mentioned before, the generation step follows the general format given in Table 6.1. The pseudo-code for replacing the lines forward_visit(e) and backward_visit(e) is given in tables 6.2 and 6.3 respectively.

A singular shredder (a, b, c), by definition, has a singular component. Hence there is an edge $e = (c \rightarrow v)$ in Adj(c) such that $HIGH1(e) = a$, $HIGH2(e) = b$ and $HIGH3(e) = \infty$.

Hence we can generate the triple when we are about to explore e. In order to find out whether the triple is degenerate or not, we need to know what q is, i.e. we need to know which child of b is currently active (in the dfs). We keep track of this information in the array active_child, which is updated whenever a recursive call is made. The rest of the section gives an informal description of how the generation step finds non-singular shredders, before giving proofs of correctness and the time bound for the generation step.

Let (a, b, c) be a non-singular shredder. By Lemma 6.2.7, $a = \text{HIGH1}(e)$, where e is as in that lemma. We will generate "candidate pairs" (a, c) and find the corresponding vertex b later. The candidate pairs will be stored in a data structure that we call PSTACK. This will be a stack of "blocks", separated by end-markers, similar to the stack in a recursion. The individual blocks will be ordered lists of candidate pairs. Before each call dfs_step(v) for a *non-leftmost vertex* v, an end-marker is inserted on top of PSTACK, signifying that a fresh block is now on top of PSTACK. After the exit from dfs_step(v), the topmost block (and the end-marker) are removed from the PSTACK. The first step of generating a triple (a, b, c) begins when some edge e in Adj(c) is explored, at which point HIGH1(e) will be our guess for a, which might be revised later. In addition, the value of α for this "candidate pair" is set to corner(e). This candidate pair (a, c) is added to the beginning of the current (topmost) block of PSTACK, and will be removed from PSTACK either to be moved to a list of triples (when the vertex b is detected), or to be discarded. As the dfs backs up over the tree path $a \rightarrow^* c$, we expect to recognize a vertex on the path as the right "b" for the pair. There are three situations in which we realize that we have come across b, corresponding to the conditions in Lemma 6.2.8:

1. Whenever we back up over a non-leftmost edge $u \rightarrow v$, we mark u as the vertex b for all the candidate pairs in the current block of PSTACK (and move them to a list of triples). In particular, the topmost block of PSTACK (and the end-marker below it) are removed.

2. Whenever we explore a non-leftmost edge e in Adj(u) and see a candidate pair (a, c) in the current block of PSTACK with $a < \text{HIGH1}(e)$, we create the triple (a, u, c).

3. Whenever we back up over a leftmost edge $u \to v$, and see a candidate pair (a, c) in the current block of PSTACK with $\text{RCH}(u) < \alpha$, we create the triple (a, u, c) and remove the pair (a, c) from PSTACK.

In situation 2 above, in addition to generating the corresponding triple, we also *revise* the value of a in the pair to HIGH1(e) (the revised pair stays on PSTACK). This corresponds to Lemma 6.2.7(ii). A pair that has already been revised once, however, is discarded. This is essential for keeping the overall time taken for the PSTACK operations linear.

Table 6.2: Generation step: pseudo-code for forward_visit(e)

```
if e = (u → v) AND HIGH3(v) = ∞ then     comment generating a singular triple
    generate the triple (a=HIGH1(v), b=HIGH2(v), c=u) and mark it non-degenerate or
    degenerate depending on whether HIGH1(q) > a or not (where q = active_child(b))
if e is non-leftmost then begin
    let (a, c) be the first candidate pair in the current block of PSTACK (set it to null
    if the end-marker is encountered instead)
    while (a, c) is not null AND a < HIGH1(e) do begin
        create the triple (a, b=u, c)
        if the pair (a, c) is unrevised then
            set a = HIGH1(e) and mark the pair (a, c) as revised
        else discard the pair (a, c) from PSTACK
        set (a, c) to the next pair in the current block (set it to null if the end-marker
        is encountered instead)
    end
    add the pair (a=HIGH1(e), c=u) to the beginning of the current block; set α =
    corner(e)
    if e is a tree edge then add an end-marker on top of PSTACK
end
if e is a tree edge then set active_child(u) = v
```

If u is the current vertex in the dfs, and (a, c) is a candidate pair in the current block, then c would be a leftmost descendant of u, and a would be an ancestor of u. Thus the pairs in the current block consist of vertices that are all on a tree path containing u. Moreover, the pairs $(a_1, c_1), (a_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, c_2, c_1, u, a_1, a_2, \ldots$ appear on this path in the order listed, and that $\ldots \leq \alpha_2 \leq \alpha_1$. This is essential for efficiently updating the pairs on PSTACK.

Theorem 6.3.1 *The generation step runs in time $O(n + m)$ and therefore the number of*

Table 6.3: Generation step: pseudo-code for backward_visit(e)

1 **if** $e = (u \to v)$ is non-leftmost **then** **comment** backing up over a non-leftmost edge
2 move all pairs in the current block to a list of triples, setting $b = u$; remove the end-marker
3 **else** **comment** backing up over a leftmost edge
4 **while** the pair (a, c) in the beginning of the current block has $\alpha > \mathrm{RCH}(u)$ **do**
5 remove the pair from PSTACK and create the triple $(a, b = u, c)$

triples generated is also $O(n + m)$.

Proof: Since the generation step has the format given in Table 6.1, we only need to verify that the **while** loops in the pseudo-code for forward_visit and backward_visit (line 5 of Table 6.2 and line 4 of Table 6.3 respectively) and line 2 of Table 6.3 take $O(n + m)$ time overall. The total number of distinct candidate pairs processed on PSTACK is at most $2m$, since each edge leads to the generation of at most one candidate pair, and this pair may be revised only once. Since the time taken by the **while** loops and line 2 of Table 6.3 is at most the number of distinct candidate pairs plus the number of edges, it follows that the overall time taken is $O(n + m)$. $\square$

Before the next lemma, we need a definition. For a vertex u, the (maximal) tree path joining the root to the unique leaf that is a leftmost descendant of u is called the *canonical path* containing u.

Lemma 6.3.2 *During the generation step, the following condition holds immediately before the* ***while*** *loops in forward_visit(e) and backward_visit(e) (line 5 of Table 6.2 and line 4 of Table 6.3 respectively). If u is the current vertex in the search, then the pairs in the current block of* PSTACK *consist of vertices that are all on the canonical path containing u. Moreover, the pairs $(a_1, c_1), (a_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, a_2, a_1, u, c_1, c_2, \ldots$ appear on this path in the order listed (possibly with repetition), and $\ldots \leq \alpha_2 \leq \alpha_1$.*

Proof: We shall proceed by induction, proving that all the operations that change the current vertex, or the current block of PSTACK (or both) preserve the properties stated in the lemma. Suppose that at some time instant in the generation step, the pairs in the current block satisfy the assertion of the lemma. Let these pairs (in order) be $(a_2, c_2), (a_3, c_3), \ldots$, and let u be the current vertex. Suppose the search is exploring a non-leftmost edge e in $\mathrm{Adj}(u)$ with $\mathrm{HIGH1}(e) = a_1$. Since the revision process (the **while** loop on line 5, Table 6.2) either discards pairs with $a < a_1$ or revises them (by setting $a = a_1$), it is clear that it preserves the required property of PSTACK. If the generation step then adds the pair $(a_1, c_1 = u)$ while exploring an edge in $\mathrm{Adj}(u)$, it is easy to see that $\alpha_2 \leq \alpha_1$, and that $a_1 \leq a_2$ because of the revision process. Thus the assertion of the lemma still holds after this pair is added. Now suppose the recursive call dfs_step(v) (that makes v the current vertex) is made after the tree edge $u \to v$ is explored (i.e. after forward_visit($u \to v$).) If $u \to v$ is non-leftmost, then a fresh block would have been started before the recursive call, so the lemma holds trivially immediately after the recursive call. If $u \to v$ is leftmost, then no candidate pair with $c = u$ is yet on PSTACK, since pairs are generated only while exploring non-leftmost edges. Hence the lemma still holds immediately after the recursive call. While backing up over a non-leftmost edge $u \to v$, the current block is removed, restoring PSTACK to its state just before the recursive call dfs_step(v), and hence the lemma still holds by induction. Finally, suppose that the search is backing up over a leftmost edge $u \to v$, with (a, c) being the first pair in the current block. It suffices to verify that a is a proper ancestor of u (unless $a = u$ is the root) to prove that the lemma still holds after the search backs up over $u \to v$. Since the pair (a, c) has survived on PSTACK till the search has backed up over to the vertex u, we claim by induction that there is no edge between $A \cup A''$ and $D(v) \setminus (A'' \cup \{c\})$, where A is defined for the pair (a, c) analogous to the definition for triples, and $A'' = [c - ND(c) + 1, \alpha)$. If there is such an edge, say between A'' and $D(v) \setminus (A'' \cup \{c\})$, it would be detected in the **while** loop in Table 6.3. If there is an edge between A and $D(v) \setminus (A'' \cup \{c\})$, then either there is a non-leftmost edge in the path $v \to^* c$ (which will be detected by line 2 in Table 6.3), or $\exists v'$ on the path $v \to^* c$, $v' \neq c$, such that $\mathrm{HIGH1}(e) > a$ for some non-leftmost edge e in $\mathrm{Adj}(v')$ (which will be detected in the **while** loop in Table 6.2.) This proves the claim and implies that a must be a proper ancestor of u (unless $a = u$ is the root), otherwise $A \cup A''$ and $D(v) \setminus (A'' \cup \{c\})$ would be (non-empty) vertex sets of components of $G \setminus \{a, c\}$. This proves the lemma. $\square$

Theorem 6.3.3 *The generation step finds a (multi-)set of triples that includes the set of* 3-*shredders of* G*. Furthermore, the non-singular triples generated have no* $\mathcal{A}$-$\mathcal{C}$ *edges, the path* $q \to^* c$ *is leftmost for all such triples, and they satisfy the properties given in Lemma 6.2.7.*

Proof: First, let (a, b, c) be a singular shredder and $D(v)$ be the vertex set of a singular component of (a, b, c). Then the triple (a, b, c) will be generated when the edge $c \to v$ is explored.

Consider a non-singular shredder (a, b, c). Let e_1 be the first edge in $\mathrm{Adj}(c)$ with $b < \mathrm{HIGH1}(e_1) \leq a$, and let $\mathrm{HIGH1}(e_1) = a_1$. (We know such an edge exists because of Lemma 6.2.6.) Now $\mathrm{HIGH1}(e_1) \leq a < \mathrm{HIGH1}(c)$ by Lemma 6.2.4; in particular, e_1 is non-leftmost. It follows that the candidate pair (a_1, c) is generated when e_1 is explored (line 12 of Table 6.2), with the correct value of α. Furthermore, if $a_1 \neq a$, then by Lemma 6.2.7, there is a vertex u on the path $q \to^* c$, $u \neq c$, such that $a = \mathrm{HIGH1}(e_2)$ for some non-leftmost edge e_2 in $\mathrm{Adj}(u)$. Subject to the above condition, choose u to be closest to c and subject to this, choose e_2 to be the earliest in $\mathrm{Adj}(u)$. Since the shredder (a, b, c) does not have any A-C or B-C edges, that candidate pair (a_1, c) remains unrevised till the edge e_2 is explored, at which point the pair will be revised with a_1 being changed to a. (The fact that the revision process reaches the pair follows from Lemma 6.3.2.) Again, since (a, b, c) has no $\mathcal{A}$-$\mathcal{C}$ edges, it follows that the pair (a, c) then stays on PSTACK until the search backs up to the vertex b. By Lemma 6.2.8, the vertex b is detected by one of the two **while** loops in tables 6.2 and 6.3, or line 2 in Table 6.3. (Again, the fact that b is detected as above follows from Lemma 6.3.2.) Thus the triple (a, b, c) will be generated.

Also, if a triple (a, b, c) is generated from a candidate pair (a, c), the fact that the candidate pair survived on PSTACK till the search backed up over the edge $b \to q$ means that there is no edge between $A \cup A''$ and $D(q) \setminus (A'' \cup \{c\})$, where $A'' = [c - ND(c) + 1, \alpha)$. (This is simply the claim proved at the end of the proof of Lemma 6.3.2.) In particular, the non-singular triples generated have no $\mathcal{A}$-$\mathcal{C}$ edges. Similarly, it follows that the path $q \to^* c$ is leftmost for all the non-singular triples generated. To see that Lemma 6.2.7 holds for the non-singular triples, note that if a triple is generated from an unrevised candidate pair, then Lemma 6.2.7(i) is satisfied, and if a triple (a, b, c) is generated from a candidate pair (a, c) that had been revised at a vertex u, then b must be a proper ancestor of u, i.e. Lemma 6.2.7(ii) is satisfied. $\square$

The next section describes some intermediate computation that needs to be done before we move on to the detection steps.

6.4 Intermediate Computation

Before the detection steps described in Section 6.5, we eliminate multiple copies of non-singular and singular triples generated by the generation step. From the proof of Theorem 6.3.3, it can be seen that for non-singular triples, among multiple copies of a triple, the one with the lowest value of α should be retained. This can be efficiently done by simultaneously sorting (using radix sort) for all vertices u, the lists of triples with $c = u$ in lexicographic order of (a, b, α), and then scanning the lists for multiple entries. Similarly, if a triple (a, b, c) is generated both as a non-singular and a singular triple, then the non-singular triple is discarded. Furthermore, non-singular triples that do not satisfy the property HIGH1$(c) > a$ (as in Lemma 6.2.4) are discarded, and triples with $\mathcal{A}$,$\mathcal{B}$ or $\mathcal{C}$ empty (i.e. $a =$ root or a is the parent of b or b is the parent of c, respectively) are discarded.

6.4.1 Computing LOW1

This section describes how to compute LOW1 for all the tree edges in linear time, using a Union-Find procedure. Let $z \hookrightarrow u$ be a back edge such that no proper descendant of u is the head of any back edge, and let $u \to v \to w \to^* z$ be the tree path from u to z. It is easy to see that for for all tree edges e along the path $v \to^* z$, LOW1$(e) = u$. If we now discard the back edge $z \hookrightarrow u$ and consider another back edge as above, we can compute, in the same way, LOW1 for other tree edges for which the quantity has not yet been computed. Proceeding in this fashion, we can compute LOW1 eventually for all tree edges. However, in order to not examine edges repeatedly (and hence spend too much time), we need to contract a tree edge once its LOW1 value is computed. It can be seen that in the situation above, *contracting* any edge on the path $v \to^* z$ keeps the LOW1 values of the remaining tree edges the same. (By contracting a tree edge $x \to y$, we mean removing the edge and the vertex y, and replacing the end y in any other (tree and back) edges with the vertex

x.) In order to pick the back edges in the manner described above, all the back edges are collected in a list, sorting the edges $z \hookrightarrow u$ in increasing order of u. The back edges are then picked from this list, in order, computing LOW1 for the relevant tree edges as above and contracting them. Note that contracting those edges does not change the head of any other back edge, and hence does not affect the ordering of the list of back edges.

The edge contraction is implemented symbolically using Union-Find. A Union-Find algorithm implements a data structure for a partition of a ground set (the set of vertices in this case). Each set has a *representative* that is used to refer to the set. The Union-Find algorithm supports two operations:

union(x,y), which forms the union of the sets containing x and y (destroying the original sets), and

find-set(x), which returns the representative of the set containing x.

As the contractions are carried out, the vertex set is dynamically represented as a partition of the original vertex set. Initially all vertices are in singleton sets. In general, the vertex represented by x is the representative of the set containing x.

Contraction of an edge $x \rightarrow y$ is done by merging the sets containing x and y, setting the representative of the merged set to that of the former set. (If the Union-Find algorithm does not implement the latter requirement, we can implement an additional function on the vertex set that maps the representative assigned by the algorithm to the one desired.) Computing the LOW1 values as above requires $O(m)$ operations (union and find-set) to be performed on a ground set of size n. Note also that the unions all correspond to edges in the dfs tree (i.e. they are all of the form union(x,y) where x and y are adjacent in the dfs tree.) Hence this is a "graphical Union-Find" where the graph is actually a tree. This special case of Union-Find can be solved in time $O(m + n)$ on a random-access machine (see [17] or [19].) The classical algorithm for Union-Find (based on "weighted-union of trees with finds executing path compression"; see [50]) performs the above Union-Find in time $O((m + n)\alpha(m + n))$, where α is the functional inverse of an Ackermann-like function.

6.4.2 The "Corner" Vertices for a Triple

The vertex α for a triple (a, b, c) was defined in such a way that the interval $[c-ND(c)+1, \alpha)$ is precisely A''. Similarly we define, for a triple (a, b, c), vertices β, γ, δ and ϵ for demarcating B'', C' and $\mathcal{D}$, as follows:

β: If there exists a child v of c with (HIGH1$(v) = a$ AND HIGH2$(v) \leq b$) OR HIGH1$(v) < a$, let v_0 be the first such v and $\beta = v_0 - ND(v_0) + 1$

otherwise $\beta = c$

γ: If there exists a child v of c such that (HIGH1$(v) = a$ AND HIGH2$(v) < b$) OR (HIGH1$(v) = a$ AND HIGH2$(v) = b$ AND HIGH3$(v) < b$) OR HIGH1$(v) < a$, let v_0 be the first such v and $\gamma = v_0 - ND(v_0) + 1$

otherwise $\gamma = c$

δ: If there exists a child v of c with HIGH1$(v) < a$, let v_0 be the first such v and $\delta = v_0 - ND(v_0) + 1$

otherwise $\delta = c$

ϵ: If there exists a child v of c with HIGH1$(v) \leq b$, let v_0 be the first such v and $\epsilon = v_0 - ND(v_0) + 1$

otherwise $\epsilon = c$

Note that $A'' = [c-ND(c)+1, \alpha)$, $B'' = ([\alpha, \beta) \cup [\delta, \epsilon))$, $C'' = ([\gamma, \delta) \cup [\epsilon, c))$,and $\mathcal{D} = [\beta, \gamma)$.

6.4.3 Computing the Corner Vertices for the Triples

For the following discussion, list(u), for all vertices u, is the list of triples (a, b, c) with $c = u$.

computing ϵ

sort list(u) in decreasing order of b $\forall u \in V(G)$

for $u \in V(G)$ **do begin**

    for tree edge $e \in$ Adj(u) **do**

        mark off triples from list(u) with $\epsilon =$ corner(e) (and remove them from the list) until a triple with $b <$ HIGH1(e) is encountered set $\epsilon = u$ for all triples left over in list(u)

end

computing δ

sort list(u) in decreasing order of a $\forall u \in V(G)$

for $u \in V(G)$ **do begin**

for tree edge $e \in \text{Adj}(u)$ **do**
mark off triples from list(u) with $\delta = \text{corner}(e)$ (and remove them from the list) until a triple with $a \leq \text{HIGH1}(e)$ is encountered

set $\delta = u$ for all triples left over in list(u)

end

computing γ

sort list(u) in decreasing lexicographic order of (a, b) $\forall u \in V(G)$

for $u \in V(G)$ **do begin**

for tree edge $e \in \text{Adj}(u)$ **do**

if $\text{HIGH3}(v) = \infty$ **then**
mark off triples from list(u) with $\gamma = \text{corner}(e)$ (and remove them from the list) until a triple with ($a = \text{HIGH1}(e)$ AND $b \leq \text{HIGH2}(e)$) OR $a < \text{HIGH1}(e)$ is encountered

else
mark off triples from list(u) with $\gamma = \text{corner}(e)$ (and remove them from the list) until a triple with ($a = \text{HIGH1}(e)$ AND $b < \text{HIGH2}(e)$) OR $a < \text{HIGH1}(e)$ is encountered

set $\gamma = u$ for all triples left over in list(u)

end

computing β

sort list(u) in decreasing lexicographic order of (a, b) $\forall u \in V(G)$

for $u \in V(G)$ **do begin**

for tree edge $e \in \text{Adj}(u)$ **do**

mark off triples from list(u) with β = corner(e) (and remove them from the list) until a triple with (a = HIGH1(e) AND b < HIGH2(e)) OR $a <$ HIGH1(e) is encountered

set $\beta = u$ for all triples left over in list(u)

end

computing α (for singular triples)

similar to computing δ; simply replace the condition $a \leq$ HIGH1(e) with $a <$ HIGH1(e).

We need to define (and compute for all the triples) an additional corner vertex to demarcate A' and B', but it can be done in a similar fashion as discussed above, and is hence omitted. Finally, in preparation for the detection steps carried out in Sections 6.5.2 and 6.5.3, we need to mark those triples in which c is not adjacent to any vertex in $\mathcal{B}$. The above happens for a triple if and only if B'' is empty (i.e. $\alpha = \beta$ and $\delta = \epsilon$) *and* there is no back-edge $c \hookrightarrow v$ with $b < v < a$. Checking the first condition for all triples is trivial. The second condition is also easily checked by looking at the first child v of c with HIGH1(v) $\leq b$ (this information can be computed for all triples along with ϵ) and then examining whether the edge e *preceding* $c \rightarrow v$ in Adj(c) has HIGH1(e) $< a$ or not.

6.5 Weeding out the Non-shredders

The generation procedure gives us a set of triples that includes the set of 3-shredders of the graph. It remains to weed out those triples that are not 3-shredders, by recognizing those that have edges between the vertex sets of the (potential) components $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. A non-singular triple is discarded as soon as such a "bad" edge is found. A singular triple, on the other hand, could have these bad edges as long as it has enough singular components to make it a 3-shredder. We detect these edges in several dfs-like steps, each step dealing with a certain type or types of these bad edges. The general idea behind all the steps is the same (except for the one described in Section 6.5.6). The types of bad edges handled by a detection step would all have either their head or tail in one of the sets B or C; so the detection of the bad edges for a given triple is carried out while the search is inside B or C, more precisely, as the search backs up over the tree paths $p \rightarrow^* b$ or $q \rightarrow^* c$ respectively.

Furthermore, the other end of each of the bad edges would be in a set of vertices that we shall call the *forbidden set*. For instance, if we are dealing with $\mathcal{A}$-$\mathcal{C}$ edges, then the bad edges would all have one end in C and the other end in $A \cup A''$. Thus the detection will be done while backing up over the path $q \rightarrow^* c$, and the forbidden set in this case will be $A \cup A''$.

The triples are stored and processed on a stack similar to the one used in the generation step. The current block of the stack would contain the triples (a, b, c) for which the current vertex u is in $p \rightarrow^* b$ or $q \rightarrow^* c$, as the case may be. We examine the back edges with head u, and for each such edge e, mark off (and remove) those triples in the current block for which the other end of e is in the forbidden set. Also, while exploring a non-leftmost tree edge e in $\mathrm{Adj}(u)$, we mark off and remove those triples with $\mathrm{HIGH1}(e)$ in the forbidden set. In order to do this step efficiently, we need to keep the forbidden sets of the triples in the current block of the stack monotone i.e. if (a_1, b_1, c_1) appears before (a_2, b_2, c_2) in the current block, then the forbidden set of (a_1, b_1, c_1) contains the forbidden set of (a_2, b_2, c_2).

The general format of the detection steps will be as given in Table 6.1, with statement A being replaced by the subroutine *sort_lists* and statement B being replaced by the subroutine *load_triples*(u). The subroutine sort_lists sorts *list*(u) for all vertices u, where list(u) is the list of triples with $c = u$ (or $b = u$ as the case may be). The subroutine load_triples(u) loads list(u) at the beginning of the current block of the stack while maintaining the property mentioned in the previous paragraph. The flowchart in Figure 6.2 describes the order in which the detection steps are applied to the triples to eventually determine the set of 3-shredders of the graph. The following subsections describe the individual steps in detail.

6.5.1 Detecting $\mathcal{A}$-$\mathcal{C}$ Edges

From Theorem 6.3.3, the non-singular triples generated do not have any $\mathcal{A}$-$\mathcal{C}$ edges. Furthermore, degenerate triples do not have any $\mathcal{A}$-$\mathcal{C}$ edges, as $\mathrm{HIGH1}(q) \leq a$. Hence this step is only required for singular, non-degenerate triples.

The forbidden set for a triple (a, b, c) is $A \cup A''$ and the detection is carried out on the

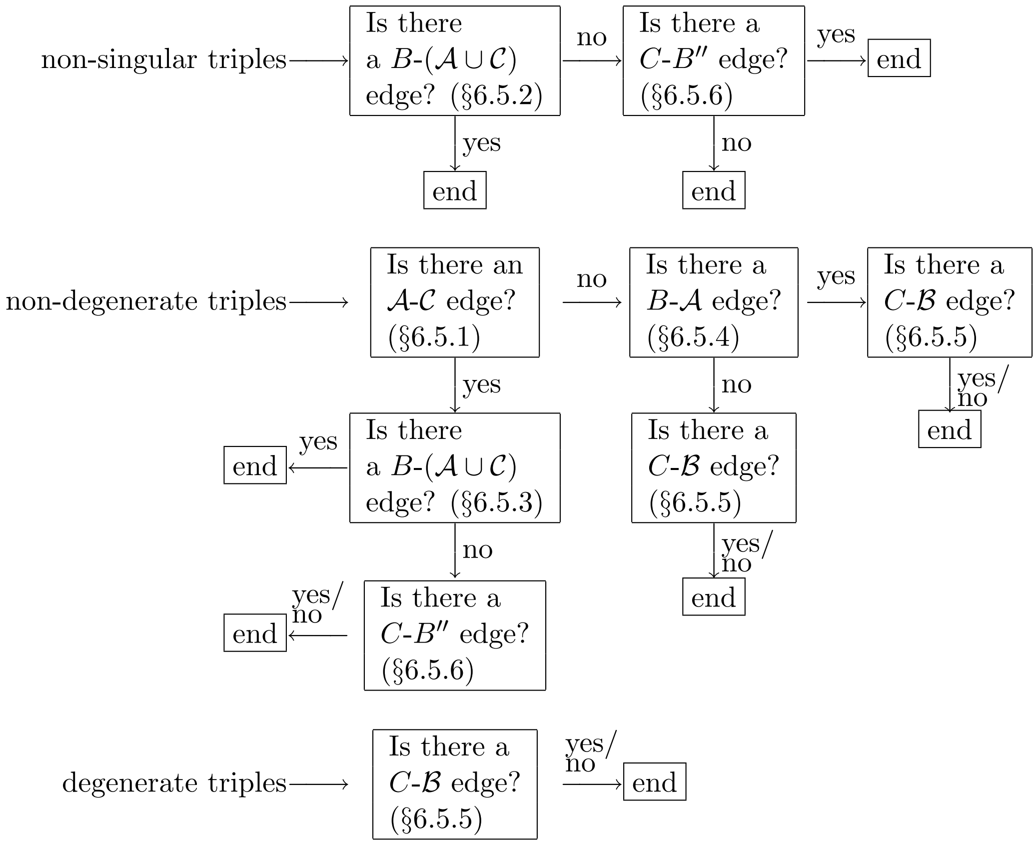

Figure 6.2: Flowchart describing the order of application of the detection steps

path $q \to^* c$. For all vertices u, list(u) is the list of non-degenerate triples with $c = u$. The subroutine sort_lists sorts list(u) $\forall u \in V$ in increasing order of a. Note that this is consistent with the monotonicity required for the forbidden sets i.e. the forbidden set $(A \cup A'')$ of (a_1, b_1, u) contains the forbidden set of (a_2, b_2, u) if $a_1 \leq a_2$. A computational remark is in order here. The above sorting can be done in linear time with a radix sort with $n + 1$ buckets, similar to the sorting of the adjacency lists described in Section 6.2. The pseudo-code for replacing forward_visit(e) and backward_visit(e) is given in tables 6.4 and 6.5 respectively. Table 6.6 gives the pseudo-code for the subroutine load_triples(u). (Recall that RCH(u), for a vertex u, is defined in Section 6.2.)

Lemma 6.5.1 *During the detection step, the following condition holds immediately before the **while** loops (line 2 of Table 6.4 and line 7 of Table 6.5). If u is the current vertex in the search, then the triples (a, b, c) in the current block of STACK are such that a, b and c*

Table 6.4: Detecting $\mathcal{A}$-$\mathcal{C}$ edges: pseudo-code for forward_visit(e)

```
if e is non-leftmost then begin
      while the triple (a, b, c) in the beginning of the current block of STACK has a <
      HIGH1(e) do
            mark the triple as having an A-C edge and remove it from STACK
      if e is a tree edge then add an end-marker on top of STACK
end
```

Table 6.5: Detecting $\mathcal{A}$-$\mathcal{C}$ edges: pseudo-code for backward_visit(e)

```
1  remove the triples with q = v from the current block of STACK
2   if e = (u → v) is non-leftmost then begin     comment  backing up over a
                                                            non-leftmost edge
3        mark all triples (a, b, c) in the current block of STACK as having an A-C edge
4        remove the block (and the end-marker) from STACK
5  end
6  else   comment backing up over a leftmost edge
7        while the triple (a, b, c) in the beginning of the current block has α > RCH(u)
         do
8              mark the triple as having an A-C edge and remove it from STACK
```

are all on the canonical path containing u, with a and b being proper ancestors of u and c a (leftmost) descendant of u. Moreover, the triples $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, a_2, a_1, u, c_1, c_2, \ldots$ appear on this path in the order listed (possibly with repetition), and that $\ldots \leq \alpha_2 \leq \alpha_1$. It follows that $(A_1 \cup A_1'') \supseteq (A_2 \cup A_2'') \ldots$ and so on.

Proof: We shall use an inductive argument similar to the one used in the proof of Lemma 6.3.2. It is clear that forward_visit(e) and the recursive call preserve the property asserted in the lemma. After backing up a non-leftmost edge $u \rightarrow v$, the property still holds by induction because STACK is restored to its state just before the recursive call dfs_step(v). After backing up a leftmost edge, the property still holds because of line 1 in Table 6.5.

Suppose now that during the search, we are at the end of dfs_step(u) and want to load list(u) on STACK. We need to verify that load_triples(u) preserves the required property of STACK. The procedure load_triples(u) looks at the triple (a_1, b_1, c_1) at the end of list(u)

Table 6.6: Detecting $\mathcal{A}$-$\mathcal{C}$ edges: pseudo-code for load_triples(u)

```
while list(u) is non-empty do begin
    let (a1, b1, c1) be at the end of list(u)
    if the current block of STACK is empty then
        remove (a1, b1, c1) from list(u) and add it at the beginning of the current block
        of STACK
    else begin
        let (a2, b2, c2) be at the beginning of the current block of STACK
        if c1 > c2 AND a1 > a2 then
            mark (a2, b2, c2) as having an A-C edge and remove it from STACK
        else remove (a1, b1, c1) from list(u) and add it at the beginning of the current
        block of STACK
    end
end
```

and compares it with the triple (a_2, b_2, c_2) at the beginning of the current block of STACK. It either moves (a_1, b_1, c_1) to STACK if it finds that doing so maintains the property stated in the lemma, or discards one of the triples, marking it as having an $\mathcal{A}$-$\mathcal{C}$ edge. (This is continued until all triples in list(u) have either been loaded onto STACK or marked as having an $\mathcal{A}$-$\mathcal{C}$ edge.) If $c_1 = c_2$ (this happens if (a_2, b_2, c_2) is originally from list(u)), then it follows that (a_1, b_1, c_1) can be moved from list(u) to STACK while maintaining the property given in the lemma. If $c_1 \neq c_2$, then c_2 is a proper leftmost descendant of c_1. Also, since (a_2, b_2, c_2) is still on STACK and the search is at the end of dfs_step(c_1), it follows that b_2 is a proper ancestor of c_1. Hence $c_1 \in C_2$. Suppose $a_1 > a_2$, and let s be the leftmost child of c_1. $D(s)$ cannot be a singular component of (a_1, b_1, c_1), since it has two distinct attachments a_2 and b_2 other than a_1. But then a singular component of (a_1, b_1, c_1) gives rise to an $\mathcal{A}$-$\mathcal{C}$ edge for (a_2, b_2, c_2), which is hence removed from STACK and marked as having an $\mathcal{A}$-$\mathcal{C}$ edge. If $a_1 \leq a_2$, it follows that (a_1, b_1, c_1) can be moved from list(u) to STACK while maintaining the property given in the lemma. $\square$

Theorem 6.5.2 *The detection step correctly marks, in $O(n + m)$ time, those of its input triples that have an $\mathcal{A}$-$\mathcal{C}$ edge.*

Proof: Suppose a triple (a, b, c) is marked by the search as having an $\mathcal{A}$-$\mathcal{C}$ edge. We need to verify that it indeed has one. If the triple is marked by the subroutine load_triples, then it can be easily seen from the proof of the previous lemma that it has an $\mathcal{A}$-$\mathcal{C}$ edge. If the triple is marked by line 3 in Table 6.4, the edge $e = u \to v$ is such that the subtree $D(v) \subseteq C$ has an attachment higher than a, that is, in A. Hence the triple has an $\mathcal{A}$-$\mathcal{C}$ edge. If the triple is marked by line 3 in Table 6.5, the edge $e = u \to v$ is non-leftmost, hence by Lemma 6.2.5 it follows that the triple has an $\mathcal{A}$-$\mathcal{C}$ edge (since $\mathrm{HIGH1}(q) > a$, by definition, for non-degenerate triples.) Finally, if the triple is marked by line 8 in Table 6.5, the vertex $u \in C$ has a back edge coming in from a vertex in A'', hence the triple has an $\mathcal{A}$-$\mathcal{C}$ edge.

Conversely, suppose a triple (a, b, c) has an $\mathcal{A}$-$\mathcal{C}$ edge. Then it has either a C-A'' edge, or a C-A edge (or both.) From the previous lemma, it then follows that in the first case, such an edge would be detected by the **while** loop on line 7 in Table 6.5. In the second case, the edge would be detected by the **while** loop in Table 6.4 or line 2 in Table 6.5. Hence a triple with an $\mathcal{A}$-$\mathcal{C}$ edge will be marked accordingly.

Finally, for the time bound, note that the subroutine load_triples takes $O(1)$ time per triple, and hence loading list(u) for all vertices u takes $O(n+m)$ time overall (as there are $O(n+m)$ triples.) Also, line 1 in Table 6.5 can be efficiently executed by maintaining, for every vertex v, the list of triples with $q = v$ (where the triple (a, b, c) is such that $a \to p \to^* b \to q \to^* c$.) The time taken for executing line 1 would then be proportional to the number of triples, and hence $O(n + m)$. $\square$

6.5.2 Detecting B-$(\mathcal{A} \cup \mathcal{C})$ Edges in Non-singular Triples

This step is only executed for non-singular triples. A triple is discarded as soon as such an edge is found. Before the detection step itself, triples (a, b, c) such that c is not adjacent to any vertex in $\mathcal{B}$ are discarded. (Note that for such triples, $\mathcal{B}$ cannot be the vertex set of a component of $G \setminus \{a, b, c\}$ by itself, and hence there is a $\mathcal{B}$-$(\mathcal{A} \cup \mathcal{C})$ edge. In fact, since c is not adjacent to any vertex in $\mathcal{B}$, such an edge must be a B-$(\mathcal{A} \cup \mathcal{C})$ edge.)

The forbidden set for a triple is $\mathcal{A} \cup \mathcal{C}$ and the detection is carried out on the path $p \to^* b$. For all vertices u, list(u) is the list of non-singular triples with $b = u$. The subroutine sort_lists sorts list(u) $\forall u \in V$ such that (a_1, u, c_1) precedes (a_2, u, c_2) in the list iff $a_1 < a_2$, or $a_1 = a_2$ and $c_1 \geq c_2$.

The pseudo-code for replacing forward_visit(e) and backward_visit(e) is given in tables 6.7 and 6.8 respectively. Table 6.9 gives the pseudo-code for the subroutine load_triples(u).

Table 6.7: Detecting B-$(\mathcal{A} \cup \mathcal{C})$ edges in non-singular triples: pseudo-code for forward_visit(e)

1 **if** e is non-leftmost **then begin**
2   **while** the triple (a, b, c) in the beginning of the current block of STACK has $a <$ HIGH1(e) **do**
3   discard the triple
4   **if** e is a tree edge **then** add an end-marker on top of STACK
5 **end**

Table 6.8: Detecting B-$(\mathcal{A} \cup \mathcal{C})$ edges in non-singular triples: pseudo-code for backward_visit(e)

1 **while** the triple (a, b, c) at the beginning of the current block of STACK has $a = u$ **do**
2   remove the triple from STACK
3 **if** $e = (u \rightarrow v)$ is non-leftmost **then**   **comment** backing up over a non-leftmost edge
4   discard all triples (a, b, c) in the current block of STACK; remove the block (and the end-marker) from STACK
5 **else** **comment** backing up over a leftmost edge
6   **for** edge $w \hookrightarrow u$ in $\mathrm{Adj}^R(u)$ **do**
7   **while** the triple (a, b, c) at the beginning of the current block has $w \in (\mathcal{A} \cup \mathcal{C})$ **do**
8   discard the triple

Lemma 6.5.3 *During the detection step, the following condition holds immediately before the **while** loops on line 2 of Table 6.7 and line 7 of Table 6.8. If u is the current vertex in the search, then the triples (a, b, c) in the current block of STACK are such that a and b are on the canonical path containing u, with a being a proper ancestor of u and b a (leftmost) descendant of u. Moreover, the triples $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, a_2, a_1, u, b_1, b_2, \ldots$ appear on this path in the order listed (possibly with repetition, and $\ldots (\mathcal{A}_2 \cup \mathcal{C}_2) \supseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$.*

Table 6.9: Detecting B-$(\mathcal{A} \cup \mathcal{C})$ edges in non-singular triples: pseudo-code for load_triples(u)

```
while list(u) is non-empty do begin
    let (a_1, b_1, c_1) be at the end of list(u)
    if the current block of STACK is empty then
        remove (a_1, b_1, c_1) from list(u) and add it at the beginning of the current block
        of STACK
    else begin
        let (a_2, b_2, c_2) be at the beginning of the current block of STACK
        if b_1 = b_2 then
            if q_1 ≠ q_2 then
                if HIGH1(q_1) > a_2 then
                    discard (a_2, b_2, c_2)
                else remove (a_1, b_1, c_1) from list(u) and add it at the beginning of the
                current block of STACK
            else if c_1 = c_2 then
                remove (a_1, b_1, c_1) from list(u) and add it at the beginning of the cur-
                rent block of STACK
            else comment c_1 > c_2
                discard (a_2, b_2, c_2)
        else if a_1 > a_2 then
            if q_1 is non-leftmost then
                discard (a_2, b_2, c_2)
            else if the second edge e in Adj(b_1) has HIGH1(e) > a_1 then
                discard (a_2, b_2, c_2)
            else if c_1 > b_2 then
                discard (a_2, b_2, c_2)
            else if c_1 = b_2 then
                if HIGH1(q_2) > a_1 then
                    discard (a_1, b_1, c_1)
                else discard (a_2, b_2, c_2)
            else if c_2 ≥ c_1 then
                discard (a_2, b_2, c_2)
            else if RCH(b_1) < α_2 then
                discard (a_2, b_2, c_2)
            else discard (a_1, b_1, c_1)
        else if c_1 ≥ b_2 OR c_1 = c_2 then
            remove (a_1, b_1, c_1) from list(u) and add it at the beginning of the current
            block of STACK
        else discard (a_2, b_2, c_2)
    end
end
```

Proof: We shall use an inductive argument similar to the one used in the proof of Lemma 6.3.2. It is clear that forward_visit(e) and the recursive call preserve the property asserted in the lemma. After backing up a non-leftmost edge $u \to v$, the property still holds by induction because STACK is restored to its state just before the recursive call dfs_step(v). After backing up a leftmost edge, the property still holds because of the **while** loop on line 1 in Table 6.8.

Suppose now that during the search, we are at the end of dfs_step(u) and want to load list(u) on STACK. The following is an outline of a proof that load_triples(u) preserves the required property of STACK.

line 10: Since c_1 is adjacent to a vertex in $\mathcal{B}_1$, this implies a B_2-A_2' edge.

line 11: In this case, $A_2 \subseteq A_1$ because of the ordering of list(u), and $(A_2' \cup D(q_2)) \subseteq A_1'$.

line 13: In this case, $A_2 \subseteq A_1$, $A_2' \subseteq A_1'$, $A_2'' \subseteq A_1''$, $C_2 = C_1$ and $C_2' \subseteq (A_1'' \cup C_1')$.

line 15: Note, in this case, that $c_1 < c_2$ cannot happen. (If $a_1 = a_2$, this follows from the ordering of list(u). If $a_1 < a_2$, then it follows because otherwise, (a_1, b_1, c_1) would never have been generated.) Since c_1 is adjacent to a vertex in $\mathcal{B}_1$, this implies a B_2-C_2 edge.

lines 16–34: If $b_1 \neq b_2$, note that b_2 is a proper leftmost descendant of b_1 and that a_2 is a proper ancestor of b_1.

line 18: $c_1 \in B_2$, hence $\mathrm{HIGH1}(c_1) > a_1$ implies a B_2-A_2 edge.

line 20: there is a B_2-A_2 edge.

line 22: Lemma 6.2.7 for (a_1, b_1, c_1) implies a B_2-A_2 edge.

line 25: Lemma 6.2.7 for (a_2, b_2, c_2) implies a B_1-A_1'' edge.

line 26: In this case, since $\mathrm{RCH}(b_1) < \alpha_1$, it follows that there is a B_2-A_2'' edge.

line 28: Note that, in this case, c_1 and c_2 may be incomparable under the ancestor relation (if q_2 is non-leftmost) or c_2 is an ancestor of c_1. In the first case, $\mathrm{RCH}(b_1) < \alpha_1$ implies a B_2-A_2' edge. In the second case, it implies a B_2-A_2'' edge.

line 30: In this case, there is a B_2-A_2'' edge.

line 31: Since $\{a_1, c_2\}$ cannot be a vertex cut, it follows that there is a vertex v in the interior of the path $a_1 \to^* c_2$ such that either $\mathrm{RCH}(v) < \alpha_2$ or there is a non-leftmost edge e in Adj(v) with $\mathrm{HIGH1}(e) > a_1$. Now v cannot be in the interior of the paths $b_2 \to^* c_2$ or

$b_1 \to^* c_1$ from Theorem 6.3.3 applied to (a_2, b_2, c_2) and (a_1, b_1, c_1) respectively. Also, it is clear that $v \neq b_1$. Hence v must be in the interior of the path $a_1 \to^* b_1$, and that implies a B_1-$\mathcal{A}_1$ edge.

line 33: Clearly, $A_2 \subseteq A_1$. If q_1 is non-leftmost, then $D(b_2) \subseteq A_1'$. Otherwise, if c_1 is an ancestor of b_2, then $(A_2' \cup D(q_2)) \subseteq A_1''$. Otherwise, if $c_1 = c_2$, then $(A_2' \cup C_2) \subseteq C_1$, $A_2'' \subseteq (A_1'' \cup C_1')$ and $C_2' \subseteq (A_1'' \cup C_1')$.

line 34: If q_2 is non-leftmost, the fact that c_1 is adjacent to a vertex in $\mathcal{B}_1$ implies a B_2-A_2' edge. Otherwise, if c_2 is a proper ancestor of c_1, it implies a B_2-A_2'' edge. Otherwise, c_1 is a proper ancestor of c_2 and the fact that c_1 is adjacent to a vertex in $\mathcal{B}_1$ implies a B_2-C_2 edge. $\square$

Theorem 6.5.4 *The detection step correctly discards, in $O(n+m)$ time, those of its input triples that have a B-$(\mathcal{A} \cup \mathcal{C})$ edge.*

Proof: Suppose a triple (a, b, c) is discarded by the search for having a B-$(\mathcal{A} \cup \mathcal{C})$ edge. We need to verify that it indeed has one. If the triple is discarded by the subroutine load_triples, then it can be easily seen from the proof of the previous lemma that it has a B-$(\mathcal{A} \cup \mathcal{C})$ edge. If the triple is discarded by line 3 in Table 6.7, the edge $e = u \to v$ is such that the subtree $D(v) \subseteq B$ has an attachment higher than a, that is, in A. Hence the triple has a B-A edge. If the triple is marked by line 4 in Table 6.8, the edge $e = u \to v$ is non-leftmost, hence by Lemma 6.2.3 it follows that the triple has a B-A edge. Finally, if the triple is marked by line 8 in Table 6.8, the vertex $u \in B$ has a back edge coming in from a vertex in $\mathcal{A} \cup \mathcal{C}$, hence the triple has a B-$(\mathcal{A} \cup \mathcal{C})$ edge.

Conversely, suppose a triple (a, b, c) has a B-$(\mathcal{A} \cup \mathcal{C})$ edge. Then it has either a B-$(A' \cup A'' \cup C \cup C')$ edge, or a B-A edge (or both.) From the previous lemma, it then follows that in the first case, such a triple would be discarded by the **while** loop on line 7 in Table 6.8. In the second case, it would be discarded by the **while** loop in Table 6.7 or line 4 in Table 6.8. Hence a triple with a B-$(\mathcal{A} \cup \mathcal{C})$ edge will be discarded.

Finally, for the time bound, note that the subroutine load_triples takes $O(1)$ time per triple, and hence loading list(u) for all vertices u takes $O(n+m)$ time overall (as there are $O(n+m)$

triples.) □

6.5.3 Detecting B-($\mathcal{A} \cup \mathcal{C}$) Edges in Singular Triples

This step is for non-degenerate triples with an $\mathcal{A}$-$\mathcal{C}$ edge. Before the detection step, the triples (a, b, c) such that c is not adjacent to any vertex in $\mathcal{B}$ are marked as having a B-($\mathcal{A} \cup \mathcal{C}$) edge, and hence not examined during the detection step. (Refer to the corresponding argument in Section 6.5.2.)

The forbidden set for a triple is $\mathcal{A} \cup \mathcal{C}$ and the detection is carried out along the path $p \rightarrow^* b$. The subroutine sort_lists sorts list(u) $\forall u \in V$ such that (a_1, u, c_1) precedes (a_2, u, c_2) in the list iff $a_1 < a_2$, or $a_1 = a_2$ and $c_1 \leq c_2$. The pseudo-code for forward_visit(e) and backward_visit(e) is similar to that in tables 6.7 and 6.8 respectively. The difference is that instead of discarding a triple, we only mark it as having a $\mathcal{B}$-($\mathcal{A} \cup \mathcal{C}$) edge and remove it from STACK. The pseudo-code for load_triples(u) is given in Table 6.10.

Lemma 6.5.5 *During the detection step, the following condition holds immediately before the* ***while*** *loops on line 2 of Table 6.7 and line 7 of Table 6.8. If u is the current vertex in the search, then the triples (a, b, c) in the current block of STACK are such that a and b are on the canonical path containing u, with a being a proper ancestor of u and b a (leftmost) descendant of u. Moreover, the triples $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, a_2, a_1, u, b_1, b_2, \ldots$ appear on this path in the order listed (possibly with repetition), and $\ldots (\mathcal{A}_2 \cup \mathcal{C}_2) \supseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$.*

Proof: We shall use an inductive argument similar to the one used in the proof of Lemma 6.3.2. It is clear that forward_visit(e) and the recursive call preserve the property asserted in the lemma. After backing up a non-leftmost edge $u \rightarrow v$, the property still holds by induction because STACK is restored to its state just before the recursive call dfs_step(v). After backing up a leftmost edge, the property still holds because of the **while** loop on line 1 in Table 6.8.

Suppose now that during the search, we are at the end of dfs_step(u) and want to load list(u) on STACK. The following is an outline of a proof that load_triples(u) preserves the required property of STACK.

line 10: Since c_1 is adjacent to a vertex in $\mathcal{B}_1$, this implies a B_2-A_2' edge.

line 11: Clearly, $A_2 \subseteq A_1$ and $A_2' \subseteq A_1'$. Since $\text{HIGH1}(q_2) > a_2 \geq a_1$, $D(q_2) \subseteq A_1'$.

line 14: Clearly, (a_1, b_1, c_1) can be moved form list(u) to STACK.

line 15: Since $(\mathcal{A}_2 \cup \mathcal{C}_2) \nsubseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$, $c_1 \in A_2''$. But since c_1 is adjacent to a vertex in $\mathcal{B}_1$, this implies a B_2-$\mathcal{A}_2$ edge.

line 17: Clearly, since $a_2 \geq a_1$, $A_2 \subseteq A_1$, $A_2' \subseteq A_1'$, $A_2'' \subseteq A_1''$ and $C_2' \subseteq (A_1'' \cup C_1')$

line 18: If c_1 and c_2 are incomparable under the ancestor relation, then $c_1 \in C_2$ and since c_1 is adjacent to a vertex in $\mathcal{B}_1$, it follows that there is a B_2-C_2 edge. On the other hand, if c_1 is a proper ancestor of c_2, then $c_1 \in C_2$ and $a_1 < a_2$ (because of the ordering of list(u).) Let s be the child of c_1 with $c_2 \in D(s)$. Clearly, $D(s) \subseteq A_1''$, and since c_1 is adjacent to a vertex in $\mathcal{B}_1$, it follows that there is a B_2-C_2 edge.

lines 19–37: If $b_1 \neq b_2$, note that b_2 is a proper leftmost descendant of b_1 and that a_2 is a proper ancestor of b_1.

line 21: If q_1 is non-leftmost, then clearly there is a B_2-A_2 edge. If c_1 is a proper ancestor of b_2, let s be the child of c_1 with $b_2 \in D(s)$. Since a singular component of (a_2, b_2, c_2) has an attachment at a_2, $D(s)$ is not a singular component of (a_1, b_1, c_1) and it follows that there is a B_2-A_2 edge. Finally, if $c_1 = b_2$ or c_1 is a descendant of b_2 not contained in $D(q_2)$, a similar argument shows that there is a B_2-A_2' edge.

line 23: It can be seen that c_2 is not contained in any singular component of (a_1, b_1, c_1). It follows that there is a B_2-C_2 edge.

line 24: It can be seen that c_1 is not contained in any singular component of (a_2, b_2, c_2). It follows that there is a $\mathcal{B}_1$-$\mathcal{C}_1$ edge.

line 26: If q_1 is non-leftmost, then $D(b_2) \subseteq A_1'$. If c_1 is an ancestor of b_2, then $(A_2' \cup D(q_2)) \subseteq A_1''$. (Note that $\text{HIGH1}(q_2) > a_2 \geq a_1$ since the triples in this step are non-degenerate.)

line 29: Clearly, (a_1, b_1, c_1) can be moved from list(u) to STACK.

line 30: This happens only when $c_1 \in A_2'$, in which case, there is a B_2-A_2' edge (as c_1 is adjacent to a vertex in $\mathcal{B}_1$.)

line 33: Clearly, (a_1, b_1, c_1) can be moved from list(u) to STACK.

line 34: In this case, $c_1 \in A_2''$, and there is a B_2-A_2'' edge (as c_1 is adjacent to a vertex in

Table 6.10: Detecting B-$(\mathcal{A} \cup \mathcal{C})$ edges in singular triples: pseudo-code for load_triples(u)

```
while list($u$) is non-empty do begin
    let $(a_1, b_1, c_1)$ be at the end of list($u$)
    if the current block of STACK is empty then
        remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the current block
        of STACK
    else begin
        let $(a_2, b_2, c_2)$ be at the beginning of the current block of STACK
        if $b_1 = b_2$ then
            if $q_1 \neq q_2$ then
                if HIGH1($q_1$) $> a_2$ then
                    mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
                else remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the
                current block of STACK
            else if $c_2$ is a proper ancestor of $c_1$ then
                if $(\mathcal{A}_2 \cup \mathcal{C}_2) \subseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$ then
                    remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the
                    current block of STACK
                else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
            else if $c_1 = c_2$ then
                remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the cur-
                rent block of STACK
            else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{C}$ edge and remove it from STACK
        else if $a_1 > a_2$ then
            if $c_1 \geq b_2$ OR $c_1 \notin D(q_2)$ then
                mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
            else if $c_1 > c_2$ then
                mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{C}$ edge and remove it from STACK
            else mark $(a_1, b_1, c_1)$ as having a $\mathcal{B}$-$\mathcal{C}$ edge and remove it from list($u$)
        else if $c_1 \geq b_2$ then
            remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the current
            block of STACK
        else if $c_1 \notin D(q_2)$ then
            if $(\mathcal{A}_2 \cup \mathcal{C}_2) \subseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$ then
                remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the cur-
                rent block of STACK
            else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
        else if $c_2$ is a proper ancestor of $c_1$ then
            if $(\mathcal{A}_2 \cup \mathcal{C}_2) \subseteq (\mathcal{A}_1 \cup \mathcal{C}_1)$ then
                remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the cur-
                rent block of STACK
            else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
        else if $c_1 = c_2$ then
            remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the current
            block of STACK
        else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{C}$ edge and remove it from STACK
    end
end
```

$\mathcal{B}_1$.)

line 36: In this case, $(A_2' \cup C_2) \subseteq C_1$, $A_2'' \subseteq A_1''$ and $C_2' \subseteq (A_1'' \cup C_1')$

line 37: It can be seen that c_2 is not in any singular component of (a_1, b_1, c_1), hence there is a B_2-C_2 edge. □

Theorem 6.5.6 *The detection step correctly detects, in $O(n+m)$ time, those of its input triples that have a B-$(\mathcal{A} \cup \mathcal{C})$ edge.*

Proof: The proof of this is nearly identical to the proof of Theorem 6.5.4 and is hence omitted.

6.5.4 Detecting B-$\mathcal{A}$ Edges

This step is for non-degenerate triples with no $\mathcal{A}$-$\mathcal{C}$ edges. The forbidden set for a triple is $\mathcal{A}$ and the detection is carried out on the path $p \rightarrow^* b$. The subroutine sort_lists sorts list(u) $\forall u \in V$ such that (a_1, u, c_1) precedes (a_2, u, c_2) in the list iff $a_1 < a_2$, or $a_1 = a_2$ and $c_1 \geq c_2$.

The pseudo-code for forward_visit(e) and backward_visit(e) is similar to that in tables 6.7 and 6.8 respectively. The difference is that instead of discarding a triple, we only mark it as having a B-$\mathcal{A}$ edge and remove it from STACK. Also, the forbidden set in this case is $\mathcal{A}$ instead of $\mathcal{A} \cup \mathcal{C}$ (see line 7 in Table 6.8.) The pseudo-code for load_triples(u) is given in Table 6.11.

Lemma 6.5.7 *During the detection step, the following condition holds immediately before the **while** loops on line 2 of Table 6.7 and line 7 of Table 6.8. If u is the current vertex in the search, then the triples (a, b, c) in the current block of STACK are such that a and b are on the canonical path containing u, with a being a proper ancestor of u and b a (leftmost) descendant of u. Moreover, the triples $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, a_2, a_1, u, b_1, b_2, \ldots$ appear on this path in the order listed (possibly with repetition), and $\ldots \mathcal{A}_2 \supseteq \mathcal{A}_1$.*

Proof: We shall use an inductive argument similar to the one used in the proof of Lemma 6.3.2. It is clear that forward_visit(e) and the recursive call preserve the property asserted in the lemma. After backing up a non-leftmost edge $u \to v$, the property still holds by induction because STACK is restored to its state just before the recursive call dfs_step(v). After backing up a leftmost edge, the property still holds because of the **while** loop on line 1 in Table 6.8.

Suppose now that during the search, we are at the end of dfs_step(u) and want to load list(u) on STACK. The following is an outline of a proof that load_triples(u) preserves the required property of STACK. Note that for non-degenerate triples with no $\mathcal{A}$-$\mathcal{C}$ edges, an argument similar to the one in Lemma 6.2.5 shows that the path $q \to^* c$ is leftmost.

line 11: Clearly, there is a B_2-A_2' edge.

line 12: Note that this happens only when $a_1 = a_2$ and LOW1$(q_1) = a_1$. Since c_1 is not adjacent to any vertex in $\mathcal{B}_1$, there is a $\mathcal{B}_1$-$(\mathcal{A}_1 \cup \mathcal{C}_1)$ edge. In fact, since LOW1$(q_1) = a_1$, there cannot be any $\mathcal{B}_1$-$\mathcal{C}_1$ edge, hence there must be a B_1-$\mathcal{A}_1$ edge.

line 13: Clearly, $A_2 \subseteq A_1$ and $(A_2' \cup D(q_2)) \subseteq A_1'$.

line 14: Suppose $c_2 > c_1$. This means that $a_2 > a_1$ (because of the order of list(u)). It follows that c_1 is not in any singular component of (a_2, b_2, c_2), which implies an $\mathcal{A}_1$-$\mathcal{C}_1$ edge, a contradiction. Hence $c_1 \geq c_2$. It can now be seen that $A_2' \subseteq A_1'$ and $A_2'' \subseteq A_1''$.

lines 15–22: If $b_1 \neq b_2$, note that b_2 is a proper leftmost descendant of b_1 and that a_2 is a proper ancestor of b_1.

line 17: If q_1 is non-leftmost, clearly there is a B_2-A_2 edge. If c_1 is a proper ancestor of b_2, it can be seen that b_2 is not contained in a singular component of (a_1, b_1, c_1), hence there is again a B_2-A_2 edge. If $c_1 = b_2$, $D(q_2)$ is not a singular component of (a_1, b_1, c_1). It follows that there is a B_2-A_2' edge.

line 19: First note that $c_1 > c_2$ is not possible. (In that case, c_1 would be a proper ancestor of c_2. Since c_2 is not contained in a singular component of (a_1, b_1, c_1), this would imply an $\mathcal{A}_2$-$\mathcal{C}_2$ edge, a contradiction.) Hence $c_2 \geq c_1$. If c_2 is an ancestor of c_1, it follows that a singular component of (a_1, b_1, c_1) implies a B_2-A_2'' edge. If not (i.e. if q_2 is non-leftmost) then a singular component of (a_1, b_1, c_1) implies a B_2-A_2' edge.

line 21: If q_1 is non-leftmost, then $D(b_2) \subseteq A_1'$. If c_1 is an ancestor of b_2, then $(A_2' \cup D(q_2)) \subseteq A_1''$. (Note that HIGH1$(q_2) > a_2 \geq a_1$ since the triples in this step are non-degenerate.)

Table 6.11: Detecting B-$\mathcal{A}$ edges: pseudo-code for load_triples(u)

```
1  while list($u$) is non-empty do begin
2      let $(a_1, b_1, c_1)$ be at the end of list($u$)
3      if the current block of STACK is empty then
4          remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the current block
           of STACK
5      else begin
6          let $(a_2, b_2, c_2)$ be at the beginning of the current block of STACK
7          if $b_1 = b_2$ then
8              if $q_1 \neq q_2$ then
9                  if HIGH1($q_1$) $> a_2$ then
10                     if LOW1($q_1$) $< a_2$ then
11                         mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from
                           STACK
12                     else mark $(a_1, b_1, c_1)$ as having a $B$-$\mathcal{A}$ edge and remove it from
                       list($u$)
13                 else remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the
                   current block of STACK
14             else remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the
               current block of STACK
15         else if $a_1 > a_2$ then
16             if $c_1 \geq b_2$ then
17                 mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
18             else    comment $c_2 \geq c_1$
19                 mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
20         else if $c_1 \geq b_2$ then
21             remove $(a_1, b_1, c_1)$ from list($u$) and add it at the beginning of the current
               block of STACK
22         else mark $(a_2, b_2, c_2)$ as having a $B$-$\mathcal{A}$ edge and remove it from STACK
23     end
24 end
```

line 22: If c_1 is an ancestor of c_2, then b_2 cannot be adjacent to any vertex in $\mathcal{A}_2$ (otherwise there would be an $\mathcal{A}_1$-$\mathcal{C}_1$ edge.) But then this means there is an $\mathcal{A}_2$-$(\mathcal{B}_2 \cup \mathcal{C}_2)$ edge, and since there cannot be a $\mathcal{A}_2$-$\mathcal{C}_2$ edge, there is in fact a B_2-$\mathcal{A}_2$ edge. On the other hand, if $c_2 > c_1$, either c_2 is a proper ancestor of c_1, or q_2 is non-leftmost. In the first case, $c_1 \in A_2''$ (since (a_2, b_2, c_2) is non-degenerate) and there is a B_2-A_2'' edge. Similarly, in the latter case, $c_1 \in A_2'$ and there is a B_2-A_2' edge. □

Theorem 6.5.8 *The detection step correctly detects, in $O(n+m)$ time, those of its input triples that have a B-$\mathcal{A}$ edge.*

Proof: The proof of this is nearly identical to the proof of Theorem 6.5.4 and is hence omitted. □

6.5.5 Detecting C-$\mathcal{B}$ Edges

This step is for non-degenerate triples with no $\mathcal{A}$-$\mathcal{C}$ edges, and also for degenerate triples, which, as observed before, do not have any $\mathcal{A}$-$\mathcal{C}$ edges either. The forbidden set for a triple is $\mathcal{A} \cup \mathcal{B}$ and the detection is carried out on the path $q \to^* c$. ($\mathcal{A}$ is included in the forbidden set only for convenience; it does not affect the detection step as it is only used for triples with no $\mathcal{A}$-$\mathcal{C}$ edges.) The subroutine sort_lists sorts list(u) $\forall u \in V$ such that (a_1, b_1, u) precedes (a_2, b_2, u) in the list iff $b_1 < b_2$, or $b_1 = b_2$ and $a_1 \leq a_2$.

The pseudo-code for replacing forward_visit(e) and backward_visit(e) is given in tables 6.12 and 6.13 respectively. Table 6.14 gives the pseudo-code for load_triples(u).

Table 6.12: Detecting C-$\mathcal{B}$ edges: pseudo-code for forward_visit(e)

if e is non-leftmost **then begin**
    while the triple (a, b, c) in the beginning of the current block of STACK has HIGH1(e) $> b$ and HIGH1(e) $\neq a$ **do**
        mark the triple as having a C-$\mathcal{B}$ edge and remove it from STACK
    if e is a tree edge **then** add an end-marker on top of STACK
end

Table 6.13: Detecting C-$\mathcal{B}$ edges: pseudo-code for backward_visit(e)

while the triple (a, b, c) at the beginning of the current block of STACK has $b = u$ **do**
    remove the triple from STACK
 if $e = (u \to v)$ is non-leftmost **then begin**   **comment** backing up over a non-leftmost edge
    mark all triples in the current block of STACK as having a C-$\mathcal{B}$ edge
    remove the block (and the end-marker) from STACK
end
else   **comment** backing up over a leftmost edge
    for edge $w \hookrightarrow u$ in Adj$^R(u)$ **do**
        while the triple (a, b, c) at the beginning of the current block has $w \in (\mathcal{A} \cup \mathcal{B})$ **do**
            mark the triple as having a C-$\mathcal{B}$ edge

Table 6.14: Detecting C-$\mathcal{B}$ edges: pseudo-code for load_triples(u)

```
while list(u) is non-empty do begin
    let (a1, b1, c1) be at the end of list(u)
    if the current block of STACK is empty then
        remove (a1, b1, c1) from list(u) and add it at the beginning of the current block
        of STACK
    else begin
        let (a2, b2, c2) be at the beginning of the current block of STACK
        if c1 = c2 then
            if b1 = b2 then
                mark (a2, b2, c2) as having a C-B edge and remove it from STACK
            else if a1 ≤ b2 OR a1 = a2 then
                remove (a1, b1, c1) from list(u) and add it at the beginning of the cur-
                rent block of STACK
            else mark (a2, b2, c2) as having a C-B edge and remove it from STACK
        else if b1 > b2 then
            mark (a2, b2, c2) as having a C-B edge
        else if b1 < b2 then
            if a1 ≤ b2 OR a1 = a2 then
                remove (a1, b1, c1) from list(u) and add it at the beginning of the cur-
                rent block of STACK
            else mark (a2, b2, c2) as having a C-B edge and remove it from STACK
        else if a1 = a2 then
            remove (a1, b1, c1) from list(u) and add it at the beginning of the current
            block of STACK
        else mark (a2, b2, c2) as having a C-B edge
    end
end
```

Lemma 6.5.9 *During the detection step, the following condition holds immediately before the* ***while*** *loops on line 2 of Table 6.12 and line 9 of Table 6.13. If u is the current vertex in the search, then the triples (a, b, c) in the current block of STACK are such that a,b and c are all on the canonical path containing u, with a and b being proper ancestors of u and c a (leftmost) descendant of u. Moreover, the triples $(a_1, b_1, c_1), (a_2, b_2, c_2), \ldots$ in the block (in order) will be such that the vertices $\ldots, b_2, b_1, u, c_1, c_2, \ldots$ appear on this path in the order listed (possibly with repetition), and $\ldots(\mathcal{A}_2 \cup \mathcal{B}_2) \supseteq (\mathcal{A}_1 \cup \mathcal{B}_1)$.*

Proof: We shall use an inductive argument similar to the one used in the proof of Lemma 6.3.2. It is clear that forward_visit(e) and the recursive call preserve the property asserted in the lemma. After backing up a non-leftmost edge $u \to v$, the property still holds by induction because STACK is restored to its state just before the recursive call dfs_step(v). After backing up a leftmost edge, the property still holds because of the **while** loop on line 1 in Table 6.13.

Suppose now that during the search, we are at the end of dfs_step(u) and want to load list(u) on STACK. The following is an outline of a proof that load_triples(u) preserves the required property of STACK.

line 9: Since $a_2 > a_1$, and there is no $\mathcal{A}_1$-$\mathcal{C}_1$ edge, a_2 cannot be adjacent to any vertex in $\mathcal{C}_2$. It follows that there must be a C_2-$\mathcal{B}_2$ edge.

line 11: If $a_1 \leq b_2$, then $(A_2 \cup B_2 \cup A_2' \cup B_2') \subseteq A_1$ and $(A_2'' \cup B_2'') \subseteq A_1''$. If $a_1 = a_2$, then $A_2 = A_1$, $(B_2 \cup A_2' \cup B_2') \subseteq B_1$, $A_2'' = A_1''$ and $B_2'' \subseteq B_1''$.

line 12: Since there is no $\mathcal{A}_2$-$\mathcal{C}_2$ edge, by considering a singular component of (a_1, b_1, c_1) it follows that $b_2 < a_1 < a_2$, and hence there is a C_2-B_2'' edge.

lines 13–21: If $c_1 \neq c_2$, note that c_2 is a proper leftmost descendant of c_1 and that b_2 is a proper ancestor of c_1.

line 14: Note that c_2 is not contained in any singular component of (a_1, b_1, c_1). Now a singular component of (a_1, b_1, c_1) must have an attachment other than b_2 and a_2, and since there is no $\mathcal{A}_2$-$\mathcal{C}_2$ edge, it follows that there must be a C_2-B_2 edge.

line 17: If $a_1 \leq b_2$, then $(A_2 \cup B_2 \cup A_2' \cup B_2') \subseteq A_1$ and $(A_2'' \cup B_2'') \subseteq D(c_2) \subseteq A_1''$. (Note that this case cannot happen for degenerate triples.) If $a_1 = a_2$, then $A_2 = A_1$ and $(B_2 \cup A_2' \cup B_2') \subseteq B_1$. Furthermore, $(A_2'' \cup B_2'') \subseteq A_1''$ in the case of non-degenerate triples, whereas $(A_2'' \cup B_2'') \subseteq B_1''$ in the case of degenerate triples.

line 18: The argument is similar to that for line 12.

line 20: In this case, A_2, B_2, A_2' and B_2' are respectively identical to A_1, B_1, A_1' and B_1'. In the case of non-degenerate triples, $(A_2'' \cup B_2'') \subseteq A_1''$. In the case of degenerate triples, $(A_2'' \cup B_2'') \subseteq B_1''$.

line 21: In this case, $a_1 < a_2$, since there is no $\mathcal{A}_2$-$\mathcal{C}_2$ edge. Now a singular component of (a_1, b_1, c_1) gives a C_2-B_2 edge. □

Theorem 6.5.10 *The detection step correctly marks, in $O(n + m)$ time, those of its input triples that have a C-$\mathcal{B}$ edge.*

Proof: Suppose a triple (a, b, c) is marked by the search as having a C-$\mathcal{B}$ edge. We need to verify that it indeed has one. If the triple is marked by the subroutine load_triples, then it can be easily seen from the proof of the previous lemma that it has a C-$\mathcal{B}$ edge. If the triple is marked by line 3 in Table 6.12, the edge $e = u \to v$ is such that the subtree $D(v) \subseteq C$ has an attachment distinct from a and higher than b, that is, in B (since the input triples for this step have no $\mathcal{A}$-$\mathcal{C}$ edges.) Hence the triple has a C-B edge. If the triple is marked by line 4 in Table 6.13, the edge $e = u \to v$ is non-leftmost, hence by Lemma 6.2.5 it follows that the triple has $\text{HIGH1}(q) \leq a$ (i.e. the input triples are degenerate.) It follows from the ordering of $\text{Adj}(u)$ that the first edge e_0 in $\text{Adj}(u)$ (or any edge before e) must have $\text{HIGH1}(e_0) = a$ and hence be a tree edge. Now if $\text{HIGH2}(e_0) \leq b$, it would mean $\text{HIGH2}(e) \leq b$, and hence c cannot be adjacent to any vertex in $\mathcal{B}$. Since the triple is degenerate, this would mean that it has a C-$\mathcal{B}$ edge. On the other hand, if $\text{HIGH2}(e_0) > b$, and $e_0 = u \to v_0$, say, then the subtree $D(v_0) \subseteq C$ has an attachment in B, so the triple has a C-B edge. Finally, if the triple is marked by line 10 in Table 6.13, the vertex $u \in C$ has a back edge coming in from a vertex in B'', hence the triple has an C-B'' edge.

Conversely, suppose a triple (a, b, c) has a C-$\mathcal{B}$ edge. Then it has either a C-B'' edge, or a C-B edge (or both.) From the previous lemma, it then follows that in the first case, such an edge would be detected by the **while** loop on line 9 in Table 6.13. In the second case, the edge would be detected by the **while** loop in Table 6.12 or line 4 in Table 6.13. Hence a triple with a C-$\mathcal{B}$ edge will be marked accordingly.

Finally, for the time bound, note that the subroutine load_triples takes $O(1)$ time per triple, and hence loading $\text{list}(u)$ for all vertices u takes $O(n+m)$ time overall (as there are $O(n+m)$ triples.) □

6.5.6 Detecting C-B'' Edges

This step is carried out for non-singular triples that do not have any B-$(\mathcal{A} \cup \mathcal{C})$ edges, and for non-degenerate triples that have an $\mathcal{A}$-$\mathcal{C}$ edge but no $\mathcal{B}$-$(\mathcal{A} \cup \mathcal{C})$ edges. For all vertices u, $\text{list}(u)$ is the list of triples, as above, with $c = u$, sorted such that (a_1, b_1, u) precedes (a_2, b_2, u) in the list iff $a_1 > a_2$, or $a_1 = a_2$ and $b_1 \leq b_2$.

For each vertex u, we first divide up list(u) into ordered *clusters* and the tree edges in Adj(u) into *sublists* corresponding to the clusters as follows. We scan list(u) in order and form clusters corresponding to (contiguous) subsequences of triples with non-decreasing values of b. It can be seen that the triples (a_1, b_1, u), (a_2, b_2, u), $\ldots$, (a_k, b_k, u) in a cluster (in order) are such that the vertices $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$ appear in the order $a_1, a_2, \ldots, a_k, b_k, \ldots, b_2, b_1$ (possibly with repetition) on the path from the root to u. We then scan Adj(u) in its usual order and make sublists of the *tree* edges in it, each sublist corresponding to a cluster in such a way that an edge e appears in the sublist corresponding to a cluster as above if $b_1 < \text{HIGH1}(e) \leq a_1$. Note that an edge can appear in at most one such sublist because the clusters do not "overlap" i.e. if we have a cluster $(a_1, b_1, u), (a_2, b_2, u), \ldots$ as above, and a cluster $(a'_1, b'_1, u), (a'_2, b'_2, u), \ldots$, then a_1,a'_1,b_1 and b'_1 cannot appear in that order (without repetition) on the path from the root to u. For non-singular triples, this is not possible because then Lemma 6.2.8 would imply that (a'_1, b'_1, u) has a B-$\mathcal{A}$ edge. Similarly, for non-degenerate triples, this is not possible because a singular component of (a_1, b_1, u) would then give rise to a B-$\mathcal{A}$ edge for (a'_1, b'_1, u).

We then sort all the sublists of Adj(u) for all vertices u in increasing order of $LOW1$. Note that for this sorting to be done in linear time overall, we use bucket sort to sort all the sublists (for all vertices u) *simultaneously*. The pseudo-code for detecting C-B'' edges is given in Table 6.15.

Table 6.15: Detecting C-B'' edges

for each vertex u **do**
    for each cluster of list(u) **do**
        let e be the first edge in the sublist of Adj(u) corresponding to the cluster and (a, b, u) be the first triple in the cluster
        while the end of the cluster or the end of the sublist is reached **do**
            if $LOW1(e) \geq b$ **then**
                set (a, b, u) to the next triple in the cluster
            else if $b < \text{HIGH1}(e) < a$ OR
                $(\text{HIGH1}(e) = a$ AND $b < \text{HIGH2}(e) < a)$ **then**
                mark the triple as having a C-B'' edge and set (a, b, u) to the next triple in the cluster
            else set e to the next edge in the sublist

Theorem 6.5.11 *The procedure in Table 6.15 detects, in $O(m+n)$ time, the triples that have a C-B'' edge.*

Proof: For a particular cluster and its corresponding sublist, the **while** loop takes as many steps as the number of triples in the cluster plus the number of edges in the sublist (plus a constant.) Since the clusters and sublists are disjoint, this implies that for a vertex u, the time taken by the inner **for** loop is linear in the size of list(u) plus the degree of u. Since there are $O(m+n)$ triples, the linear time bound follows.

Consider a triple (a, b, u) in list(u). If it is marked by the procedure as having a C-B'' edge, then it follows from the condition in line 7 of Table 6.15 that the triple indeed has a C-B'' edge. On the other hand, suppose the triple has such an edge. We need to show that it is marked so by the procedure. Let $e = (u \to v)$ be the first edge in the sublist corresponding to the cluster containing the triple such that $D(v) \in B''$ and $D(v)$ has an attachment in C. Then either $b < \text{HIGH1}(e) < a$, or $\text{HIGH1}(e) = a$ and $b < \text{HIGH2}(e) < a$. Since the sublist is ordered in increasing order of $LOW1$, the triple is still being considered (or waiting to be considered) when the edge e is being examined by the procedure. Furthermore, the triple is then marked before the procedure moves to the next edge in the sublist. This is because, for a triple occurring before (a, b, u) in the cluster, one of the conditions on lines 5 or 7 must hold, and hence the last option in the **if** statement (line 9) cannot happen. In other words, before the procedure moves to the next edge (after e) in the sublist, the triple (a, b, u) will be considered, and hence will be marked as having a C-B'' edge. This proves the correctness of the procedure, and hence proves the theorem. $\square$

REFERENCES

[1] Altshuler, A., "Construction and enumeration of regular maps on the torus," *Discrete Math.*, vol. 4, pp. 201–217, 1973. 3.5

[2] Archdeacon, D., *A Kuratowski Theorem for the Projective Plane*. PhD thesis, The Ohio State University, 1980. 2.8

[3] Archdeacon, D., "A Kuratowski theorem for the projective plane," *J. Graph Theory*, vol. 5, no. 3, pp. 243–246, 1981. 2.8, 2.8.1

[4] Archdeacon, D. and Huneke, P., "A Kuratowski theorem for nonorientable surfaces," *J. Combin. Theory Ser. B*, vol. 46, no. 2, pp. 173–231, 1989. 1.1.1

[5] Bodendiek, R. and Wagner, K., "Solution to König's graph embedding problem," *Math. Nachr.*, vol. 140, pp. 251–272, 1989. 1.1.1

[6] Bodlaender, H. L., "Treewidth: Algorithmic techniques and results," in *MFCS 1997 (Bratislava)*, vol. 1295 of *Lecture Notes in Comput. Sci.*, pp. 19–36, Berlin: Springer, 1997. 1.1.1

[7] Brualdi, R. A. and Shader, B. L., "Matrices of sign-solvable linear systems," *Cambridge Tracts in Mathematics*, vol. 116, 1995. 1.2.5

[8] Cheriyan, J. and Thurimella, R., "Fast algorithms for k-shredders and k-node connectivity augmentation," *Journal of Algorithms*, vol. 33, no. 1, pp. 15–50, 1999. 6.1

[9] Czumaj, A., Halldórsson, M. M., Lingas, A., and Nilsson, J., "Approximation algorithms for optimization problems in graphs with superlogarithmic treewidth," *Inform. Process. Lett.*, vol. 94, no. 2, pp. 49–53, 2005. 1.1.1

[10] Diestel, R., *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics*, ch. Minors, Trees, and WQO. Springer-Verlag, Heidelberg, 3 ed., 2005. 1.1.1

[11] Ding, G., Oporowski, B., Thomas, R., and Vertigan, D., "Large 4-connected non-planar graphs." In Preparation. 1.2.1, 2.1

[12] Duke, R. A. and Grünbaum, B. Personal Communication. 5.1, 5.1.1

[13] Fijavž, G., "Minor-minimal 6-connected toroidal graphs." Submitted for publication. 3.5

[14] Fijavž, G., "Minor-minimal 6-regular graphs in the Klein bottle," *European J. Combin.*, vol. 25, no. 6, pp. 893–898, 2004. 3.5

[15] Fijavž, G. and Mohar, B., "K_6-minors in projective planar graphs," *Combinatorica*, vol. 23, no. 3, pp. 453–465, 2003. 3.5

[16] Gabow, H. N., "Using expander graphs to find vertex connectivity," in *41st Annual Symposium on Foundations of Computer Science*, pp. 410–420, IEEE Comput. Soc. Press, 2000. 6.1

[17] Gabow, H. N. and Tarjan, R. E., "A linear-time algorithm for a special case of disjoint set union," *J. Comput. System Sci.*, vol. 30, no. 2, pp. 209–221, 1985. 6.4.1

[18] Grohe, M., "Local tree-width, excluded minors, and approximation algorithms," *Combinatorica*, vol. 23, no. 4, pp. 613–632, 2003. 1.1.1

[19] Gustedt, J., "Efficient Union-Find for planar graphs and other sparse graph classes," *Theoretical Computer Science*, vol. 203, no. 1, pp. 123–141, 1998. 6.4.1

[20] Gutwenger, C. and Mutzel, P., "A linear time implementation of SPQR-trees," in *Graph Drawing*, vol. 1984 of *Lecture Notes in Computer Science*, pp. 77–90, 2000. 6.1, 6.2

[21] Henzinger, M. R., Rao, S., and Gabow, H. N., "Computing vertex connectivity: New bounds from old techniques," *Journal of Algorithms*, vol. 34, no. 2, pp. 222–250, 2000. FOCS 96.

[22] Hopcroft, J. E. and Tarjan, R. E., "Dividing a graph into triconnected components," *SIAM J. Comput.*, vol. 2, no. 3, pp. 135–158, 1973. 6.1, 6.2, 6.2

[23] Johnson, T. and Thomas, R., "A splitter theorem for internally 4-connected graphs." Manuscript. 2.4

[24] Jorgensen, L., "Contractions to K_8," *Journal of Graph Theory*, vol. 18, no. 5, pp. 431–448, 1994. 1.2.1

[25] Kanevsky, A., Tamassia, R., DiBattista, G., and Chen, J., "On-line maintenance of the four-connected components of a graph," in *32nd Annual Symposium on Foundations of Computer Science*, pp. 793–801, 1991. 6.1

[26] Kastelyn, P. W., "Dimer statistics and phase transitions," *J. Math. Phys.*, vol. 4, pp. 287–293, 1963. 1.2.5

[27] Kastelyn, P. W., "Graph theory and crystal physics," in *Graph Theory and Theoretical Physics* (Harary, F., ed.), pp. 43–110, New York: Academic Press, 1967. 1.2.5

[28] Klee, V., Ladner, R., and Manber, R., "Sign-solvability revisited," *Linear Algebra Appl.*, vol. 59, pp. 131–158, 1984. 1.2.5

[29] Kostochka, A. V., "The minimum Hadwiger number for graphs with a given mean degree of vertices," *Metody Diskret. Analiz.*, no. 38, pp. 37–58, 1982. 1.1.1

[30] Kuratowski, K., "Sur le problème des courbes gauches en topologie," *Fund. Math.*, vol. 15, pp. 271–283, 1930. 1.1.1

[31] Lavrenchenko, S. A., "Irreducible triangulations of the torus," *Ukrainskiĭ Geometricheskiĭ Sbornik*, no. 30, pp. 52–62, ii, 1987. Translation in J. Soviet Math. **51** (1990), no. 5, 2537–2543. 5.1, 5.1.1

[32] Mader, W., "Homomorphiesätze für Graphen," *Math. Ann.*, vol. 178, pp. 154–168, 1968. 1.2.2

[33] Mader, W., "Über trennende Eckenmengen in homomorphiekritische Graphen," *Math. Ann.*, vol. 175, pp. 245–252, 1968. 1.2.2

[34] McCuaig, W., Robertson, N., Seymour, P. D., and Thomas, R., "Permanents, Pfaffian orientations, and even directed circuits," in *Proceedings of the Symposium on the Theory of Computing*, 1997. 1.2.5

[35] McCuaig, W., "Pólya's permanent problem," *Electron. J. Combin.*, vol. 11, no. 1, pp. Research Paper 79, 83 pp. (electronic), 2004. 1.2.5

[36] Negami, S., "Uniqueness and faithfulness of embedding of toroidal graphs," *Discrete Math.*, vol. 44, no. 2, pp. 161–180, 1983. 3.5

[37] Negami, S., "Classification of 6-regular Klein-bottlal graphs," Res. Rep. Inf. Sci. A-96, Tokyo Institute of Technology, 1984. 3.5

[38] Pólya, G., "Aufgabe 424," *Arch. Math. Phys. Ser.*, vol. 20, 1913. 1.2.5

[39] Quine, W. V. and Ullian, J. S., *The Web of Belief*, p. 133. New York: Random House, 2 ed., 1978.

[40] Reed, B. A., "Algorithmic aspects of tree width," in *Recent Advances in Algorithms and Combinatorics* (Reed, B. A. and Sales, C. L., eds.), vol. 11 of *CMS Books in Mathematics (Ouvrages de Mathématiques de la SMC)*, pp. 85–107, New York: Springer-Verlag, 2003. 1.1.1

[41] Robertson, N., Seymour, P. D., and Thomas, R., "Non-planar extensions of planar graphs." Manuscript. 1.2.1, 2.1, 2.1, 2.3, 2.3.1

[42] Robertson, N., Seymour, P. D., and Thomas, R., "A survey of linkless embeddings," in *Graph Structure Theory*, vol. 147 of *Contemporary Mathematics*, pp. 125–136, American Mathematical Society, 1993.

[43] Robertson, N., Seymour, P. D., and Thomas, R., "Permanents, Pfaffian orientations, and even directed circuits," *Ann. of Math. (2)*, vol. 150, pp. 929–975, 1999. 1.2.5

[44] Robertson, N. and Seymour, P. D., "Graph minors. VIII. A Kuratowski theorem for general surfaces," *J. Combin. Theory Ser. B*, vol. 48, no. 2, pp. 255–288, 1990. 1.1.1

[45] Robertson, N., Seymour, P., and Thomas, R., "Hadwiger's conjecture for K_6-free graphs," *Combinatorica*, vol. 13, no. 3, pp. 279–361, 1993. 1.1.1, 1.2.2

[46] Robertson, N. and Thomas, R., "On the orientable genus of graphs embedded in the Klein bottle," *J. Graph Theory*, vol. 15, no. 4, pp. 407–419, 1991. 5.2

[47] Robertson, N. and Vitray, R., "Representativity of surface embeddings," in *Paths, Flows, and VLSI-layout (Bonn, 1988)*, vol. 9 of *Algorithms Combin.*, pp. 293–328, Berlin: Springer, 1990. 5.2

[48] Seymour, P. D., "Disjoint paths in graphs," *Discrete Math.*, vol. 29, pp. 293–309, 1980. 1.1.1, 3.1.2

[49] Shiloach, Y., "A polynomial solution to the undirected *Two Paths* problem," *J. Assoc. Comput. Mach.*, vol. 27, no. 3, pp. 445–456, 1980. 1.1.1, 3.1.2

[50] Tarjan, R. E., "Efficiency of a good but not linear Set Union algorithm," *J. Assoc. Comput. Mach.*, vol. 22, pp. 215–225, 1975. 6.4.1

[51] Thomas, R. and Thomson, J., "Excluding minors in non-planar graphs of girth five," *Combinatorics, Probability and Computing*, vol. 9, pp. 573–585, 2000. 1.2.1, 2.1

[52] Thomas, R., "Recent excluded minor theorems for graphs," in *Surveys in combinatorics, 1999 (Canterbury)*, vol. 267 of *London Math. Soc. Lecture Note Ser.*, pp. 201–222, Cambridge Univ. Press, 1999. 1.2.3

[53] Thomason, A., "An extremal function for contractions of graphs," *Math. Proc. Cambridge Philos. Soc.*, vol. 95, no. 2, pp. 261–265, 1984. 1.1.1

[54] Thomassen, C., "2-linked graphs," *European J. Combin.*, vol. 1, no. 4, pp. 371–378, 1980. 1.1.1, 3.1.2

[55] Thomassen, C., "Tilings of the torus and the Klein bottle and vertex-transitive graphs on a fixed surface," *Trans. Amer. Math. Soc.*, vol. 323, no. 2, pp. 605–635, 1991. 3.5

[56] Thomassen, C., "A simpler proof of the excluded minor theorem for higher surfaces," *J. Combin. Theory Ser. B*, vol. 70, no. 2, pp. 306–311, 1997. 1.1.1

[57] Tutte, W. T., "How to draw a graph," *Proc. London Math. Soc. (3)*, vol. 13, pp. 743–767, 1963. 2.1

[58] Vazirani, V. and Yannakakis, M., "Pfaffian orientations, 0-1 permanents, and even cycles in directed graphs," *Discrete Appl. Math.*, vol. 25, pp. 179–190, 1989. 1.2.5

[59] Wagner, K., "Über eine Eigenschaft der ebenen Komplexe," *Math. Ann.*, no. 114, pp. 570–590, 1937. 1.1.1

[60] Whitney, H., "Congruent graphs and the connectivity of graphs," *Amer. J. Math.*, vol. 54, no. 1, pp. 150–168, 1932. 2.1